INL/EXT-13-30345

Exploratory Nuclear Reactor Safety Analysis and Visualization via Integrated Topological and Geometric Techniques

Dan Maljovec
Bei Wang
Diego Mandelli
Peer-Timo Bremer
Valerio Pascucci
Michael Pernice
Robert Nourgaliev

October 2013

Idaho National Laboratory

The INL is a U.S. Department of Energy National Laboratory
operated by Battelle Energy Alliance

INL/EXT-13-30345

Exploratory Nuclear Reactor Safety Analysis and Visualization via Integrated Topological and Geometric Techniques

Dan Maljovec[1]
Bei Wang[1]
Diego Mandelli
Peer-Timo Bremer[2]
Valerio Pascucci[1]
Michael Pernice
Robert Nourgaliev

[1]University of Utah
[2]Lawrence Livermore National Laboratory

October 2013

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

Prepared for the
U.S. Department of Energy
Through the INL LDRD Program
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

Issued by the Idaho National Laboratory, operated for the United States Department of Energy by Battelle Energy Alliance.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

INL/EXT-13-30345
Unlimited Release
Printed October 22, 2013

Exploratory Nuclear Reactor Safety Analysis and Visualization via Integrated Topological and Geometric Techniques

Dan Maljovec, Bei Wang, Valerio Pascucci
Scientific Computing and Imaging Institute
University of Utah
72 S Central Campus Drive
Salt Lake City, UT 84112
maljovec@cs.utah.edu, beiwang@sci.utah.edu, pascucci@sci.utah.edu

Peer-Timo Bremer
Lawrence Livermore National Laboratory
P.O. Box 808
Livermore, CA 94551-0808
bremer5@llnl.gov

Diego Mandelli, Michael Pernice, Robert Nourgaliev
Idaho National Laboratory
P.O. Box 1625
Idaho Falls, ID 83415-3840
diego.mandelli@inl.gov, michael.pernice@inl.gov, robert.nourgaliev@inl.gov

Acknowledgment

This work was supported in part by Idaho National Laboratory LDRD grant, 00115847A1 -DEAC 0705ID14517, and Enabling Transformational Science grant UTA09000731. This work was also supported in part by NSF OCI- 0906379, NSF OCI- 0904631, DOE/NEUP 120341, DOE/MAPD DESC000192, DOE/LLNL B597476, DOE/Codesign P01180734, and DOE/SciDAC DESC0-007446. This work was funded by the Uncertainty Quantification Strategic Initiative Laboratory Directed Research and Development Project at LLNL under project Tracking code 10-SI-023.LLNL-CONF-641030. It was also performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Contents

Figures

Tables

Summary

In nuclear engineering, a recent trend is the implementation of computational-heavy and time-consuming algorithms and codes for the design and safety analysis of nuclear power plants. The new generation of system analysis codes aim to embrace several phenomena such as thermo-hydraulic, structural behavior, system dynamics and human behavior, as well as uncertainty quantification and sensitivity analysis associated with these phenomena. In particular, the use of dynamic Probabilistic Risk Assessment (PRA) methodologies allows a systematic approach to uncertainty quantification. Dynamic methodologies in PRA account for possible coupling between triggered or stochastic events through explicit consideration of the time element in system evolution, often through the use of dynamic system models (e.g. simulators). They are usually needed when the system contains multiple failure modes, control loops, hardware/software, processes, or human interactions. Dynamic methodologies are also capable of modeling the consequences of epistemic and aleatory uncertainties.

The Monte-Carlo (MC) and the Dynamic Event Tree (DET) approaches belong to this new class of dynamic PRA methodologies. The major challenges in using MC and DET (as well as other dynamic methodologies) are the heavier computational and memory requirements compared to the classical Event Tree (ET) analysis. This is due to the fact that each branch generated in the event tree can contain time evolutions of a large number of variables (e.g. about 50,000 data channels are typically present in RELAP), and a large number of scenarios can be generated from a single initiating event (possibly in the order of hundreds or even thousands). Such large amounts of information are usually very difficult to analyze in order to identify the main trends in scenario evolutions and the key risk contributors for the initiating events.

This report aims to improve the Dynamic PRA methodologies by tackling the above challenges using (a) adaptive sampling techniques to reduce computational cost of the analysis and (b) topology-based methodologies to interactively visualize high-dimensional data and extract risk-informed insights. First, we employ adaptive sampling techniques that aim to predict simulation outcomes from a small number of known observations, and to decide the sample locations that potentially maximize information gain of the model. These techniques are used to explore as well as to exploit the simulation space. They are especially useful in safety analysis to focus samples along the limit surface, i.e. the boundaries in the simulation space between system failure and system success. Second, we present a software tool for the analysis and visualization of large-scale high-dimensional nuclear simulation data. We model these datasets as high-dimensional scalar functions defined over discrete samples of the domain. We provide structural analysis of such functions at multiple scales and provide insight into the relationship between the input parameters and the output. We enable exploratory analysis for users, where we help the users to differentiate features from noise through multi-scale analysis on an interactive platform, based on domain knowledge and data characterization. Our analysis is performed by exploiting the topological and geometric properties of the domain, building statistical models based on its topological segmentations and providing interactive visual interfaces to facilitate such explorations.

1 Introduction

State of the art Probabilistic Risk Assessment (PRA) methodologies largely employ static logic binary structures such as event trees (ETs) and fault trees (FTs). These tools, however, suffer several limitations: Timing of events is not explicitly accounted for; The ordering of events is preset by the analysts; The modeling of complex accident temporal behaviors is not considered and it is mainly driven by expert judgment. These issues are particularly relevant when uncertainty quantification on timing or sequencing of events is performed in order to gain risk-informed insights. Since the early 90's there has been a growing interest in methodologies that would address the limitation listed above. These new set of dynamic methodologies, known as Dynamic PRA (DPRA), explicitly consider the time variable and system dynamics. They largely employ system simulator codes to accurately model system dynamics. Typically, the system simulator codes (e.g., RELAP5 [61] or MELCOR [28]) are coupled with other codes that monitor and control the simulation (e.g., ADAPT [34], RAVEN [2]). The latter, in particular, introduce both deterministic (e.g., system control logic, operating procedures) and stochastic (e.g., component failures, variable uncertainties) elements into the simulation.

A typical DPRA scenario is performed by: (a) Sampling values of input parameters from the uncertain space of interest; (b) Simulating the system behavior for a specific set of parameter values; (c) Analyzing the set of simulation runs; (d) Visualizing the correlations between parameter values and simulation outcomes. Here, sampling from the parameter space is typically performed by randomly sampling from a given distribution (e.g., Monte-Carlo) or selecting parameter values based on user input (e.g., Dynamic Event Tree (DET) [1]). A simulation run is then performed using the sampled parameter values, which typically affect the timing and sequencing of events that occur during the simulation. Finally, the analysis focuses on identifying and visualizing the correlations between timing and sequencing of events with simulation outcomes.

In a classical (ET or FT based) PRA environment, such analysis is performed by simply observing and ranking the minimal cut sets that contribute to a Top Event (e.g., core damage). In a DPRA environment, however, the data generated is more heterogeneous since it consists of both temporal profiles of state variables and timing of specific events. The visual exploration of such data is a new research area and is especially relevant when uncertainty quantification is performed on multiple parameters for complex systems such as nuclear power plants. Such exploration aims to evaluate impact of uncertainties on simulation outcome (e.g., maximum core temperature).

This report presents state-of-the-art topology-inspired algorithms that have been developed in the past few years to improve the capabilities of DPRA methodologies. It focuses on three aspects of the above process: data generation, analysis and visualization.

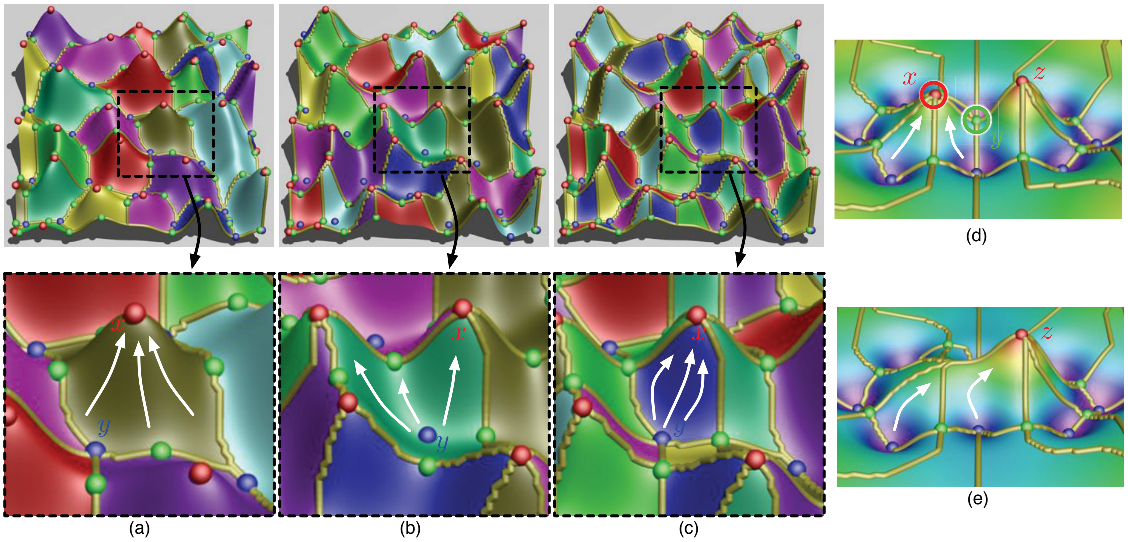
First, adaptive sampling algorithms have been developed to minimize the number of simulation runs. Nuclear simulations are in fact often computationally expensive, time-consuming, and high-dimensional with respect to the number of input parameters. Thus, exploring the space of all possible simulation outcomes is infeasible using finite computing resources. This is a typical context for performing adaptive sampling where a few observations are obtained from the simulation, a surrogate model is built in order predict behavior of the system, and new samples are selected based on the model constructed. The surrogate model is then updated based on the simulation results of the sampled points. In this way, we attempt to gain the most information possible with a small number of carefully selected sample points, limiting the number of expensive trials needed to understand features of the simulation space. From a safety point of view, we are interested in identifying the limit surface, i.e., the boundaries in the simulation space between system failure and system success.

Second, the ability to analyze and identify correlations among timing of events through system dynamics, software or human interactions is essential for nuclear power plant safety analysis in order to gain risk-informed insights. A first approach toward discovering these correlations from data generated by DPRA methodologies has been developed using Fuzzy classification. We have developed advanced topology-based clustering algorithms which are able to fully analyze these correlations by considering both the complete system dynamics and the simulation outcome.

Finally, the ability to interactively visualize the analysis results could greatly assist the user in identifying system characteristics for simulated accident scenarios. We present a software tool that enriches the DPRA data exploratory experience with analysis and visualization capabilities. We demonstrate the versatility of our tool through various nuclear datasets.

2 Technical Background

Our topology-based techniques rely heavily on the concepts of Morse-Smale complex, its approximation in high dimension and persistence simplification. We provide minimal technical background on these concepts with intuitive examples, to convey a basic understanding of these concepts for non-specialists. It is important to note, that topological clustering assumes we can treat our data as a scalar function, where an arbitrary number of inputs in the the domain space, map to scalar output values in the range space. The clustering we provide, defined in terms of gradient behavior with respect to the output value, maintains a coherent, non-overlapping segmentation of the domain space.



(a) (b) (c) (d) (e)

Figure 1: For a height function defined on a 2D domain (where maxima, minima and saddles are colored red, blue and green respectively): (a) For each descending manifold, the gradient flow (white arrow) ends a the same maxima; (b) For each ascending manifold, the gradient flow starts at the same minimum; (c) For each Morse-Smale crystal, the gradient flow begins and ends at the same maximum-minimum pair. To illustrate persistence simplification: In (d), the left peak at the maximum x is considered less important topologically than its nearby peak at maximum z, since x is lower. Therefore, at a certain scale, we would like to represent this feature as a single peak instead of two separate peaks, as shown in (e), by redirecting gradient flow (white arrow) that originally terminates at x to terminate at z. In this way, we simplify the function by removing (canceling) the local maximum x with its nearby saddle y.

Morse-Smale Complex and Its Approximation. The foundation of our analysis and visualization tool, as well as the adaptive sampling and clustering methodology, is based upon the topological structure know as the Morse-Smale complex [22, 23, 33], which is derived from the Morse theory [57]. Let $f : \mathbb{M} \to \mathbb{R}$ be a smooth function defined on a smooth manifold embedded in $\mathbb{R}^n$. A point $x \in \mathbb{M}$ is called *critical* if its gradient (e.g. a vector that points in the direction of the greatest increase of the function) $\vec{\nabla} f(x) = 0$, otherwise it is *regular*. At any regular point x the gradient is well-defined and integrating it traces out an integral line. The function increases along the integral line, which begins at a local minimum and ends at a local maximum of f. The *ascending/descending* manifold (or unstable/stable manifold) of a critical point p is defined as all points whose integral lines start/end at p. The descending manifolds form the *Morse complex* of f while the ascending manifolds form the *Morse complex* of $-f$. The set of intersections of ascending and descending manifolds creates the *Morse-Smale* complex of f. Each *crystal* of the Morse-Smale complex is a union of integral lines that all share the same origin and the same destination. In other words, all the points inside a single crystal have uniform gradient flow behavior. These crystals yield a decomposition into monotonic, non-overlapping regions of the domain. Figure 1(a)-(c) illustrate these concepts.

To approximate the Morse-Smale complex of a high-dimensional scalar function f defined on a finite set of points $\mathbb{X}$ in $\mathbb{R}^n$, we would need to estimate the gradient at each input point. This is done by imposing a neighborhood graph that connects points in $\mathbb{X}$ with edges that approximates its underlying structure. At each point in $\mathbb{X}$, we choose the steepest ascending edge to represent the gradient. With this gradient approximation, we can determine the extrema by labeling all points with no neighbors of higher values as maximum and all points with no neighbors of lower values as minimum. We then label all points in $\mathbb{X}$ according to the local extrema at which its gradient begins and terminates. Subsequently, we collect all vertices with the same pair of labels into crystals and add the extrema to all crystals that share the corresponding label. These crystals form an approximation of the Morse-Smale complex [29]. The key component in our proposed work is that such Morse-Smale crystals form the clustering of $\mathbb{X}$. In fact, such type of clustering based on topological structure is not entirely new. As pointed out in [29], the Morse complex can be thought of as a variant of mean shift clustering in which the kernel density estimation is replaced with function values of f.

Figure 2 illustrates how our approximation works for a 2D height function where the function values are only given at a finite set of sampled points in the domain. The graph shown is part of a k-nearest neighbor graph where $k = 4$. Point x (y) is a local maximum (local minimum) since it has no neighbors in the graph with higher (lower) function values. By following the approximated gradients, all the cyan points belong to the same crystal since they share the same pairs of maximum-minimum labels, e.g. (x, y).

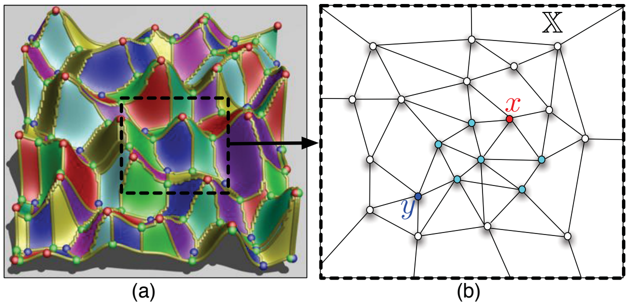


Figure 2: (a): The Morse-Smale complex of a 2D height function f. (b): The approximation of a Morse-Smale crystal of f, where the values of f are only given at a finite set of sampled points in the 2D domain.

Neighborhood graphs. While many neighborhood graphs are possible [16], we use a *relaxed Gabriel graph* [5] here which has certain desirable properties (such as relatively dense connectivities). The Gabriel graph [27] is a specific type of β-skeleton graph ($\beta = 1$) all of which belong to a larger class of graphs known as empty region graphs. As such, these class of graphs are com-

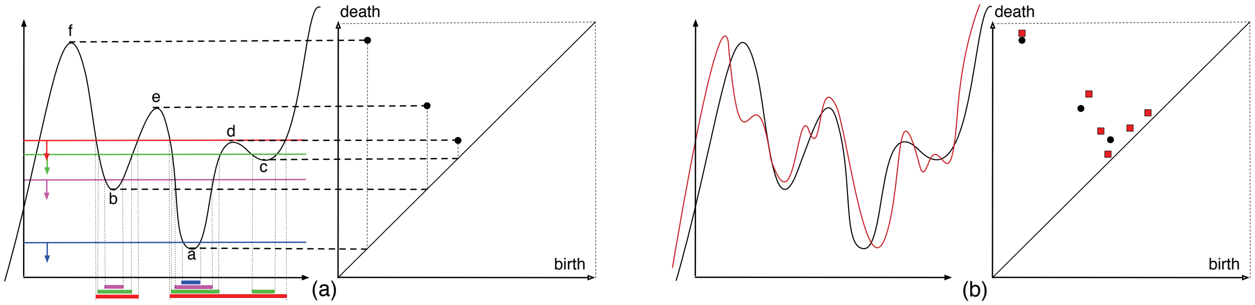


Figure 3: (a) A 1D function with three local minima and three local maxima. The critical points are paired, and each pair is encoded as a point in the persistence diagram on the right. (b) Left, two 1D functions f and g with small L_∞-distance. Right, their corresponding persistence diagrams of f (circles) and g (squares) have small bottleneck distance.

prised of all edges that pass an empty region test. For the Gabriel graph, this entails circumscribing a proposed edge between points p and q with a d-dimensional ball with a diameter equal to the length of the edge $\overline{pq}$. As long as no other points lie within the circumscribing ball, the edge is added to the graph. We can relax this property by further prescribing that only neighbors of p and q are considered in the empty region test. Thus, if r is not a neighbor of p or q, but lies within the circumscribing region, then the edge $\overline{pq}$ may be added. In order for this process to work correctly, we begin at a given point and add edges in increasing order of distance.

Persistence and Persistence Simplification. In real datasets, there is often noise which may manifest itself as small topological artifacts, either spurious extrema that may not truly exist in the data or small features the user does not deem relevant. To account for this and allow the user to select a *scale* appropriate for the specified dataset, we use the notion of *persistence simplification* [21, 24], whereby less salient features are merged with neighboring, more significant features. We illustrate the persistence simplification procedure intuitively through the example in Figure 1(d)-(e).

The theory of persistence was first introduced in [10, 24], but borrows from the conventional notion of the saliency of watersheds in image segmentation. It has since been applied to a number of problems, including sensor networks [18], surface description and reconstruction [8], protein shapes [21], images analysis [9], and topological de-noising [42]. In visualization, it has been used to simplify Morse-Smale complexes [6, 32], Reeb graphs [15] and contour trees [11]. Here we introduce persistence for a 1D (single variable) function [21] and refer to [10, 13, 24] for its general settings.

For a one-dimensional smooth function $f : \mathbb{R} \to \mathbb{R}$, persistence can be described through the number of connected components in the sublevel sets, and by tracking the birth and death of these components. In particular, components are created and destroyed only at sublevel sets containing critical points. Pairing the critical point that creates a component with the one that destroys it thus creates a pairing of critical points. Suppose f has non-degenerate critical points with distinct function values. We have two types of critical points, (local) maxima and (local) minima. We consider the sublevel sets of f, $F_t = f^{-1}(-\infty, t]$ and track the connectivity of F_t as we increase t from $-\infty$. As shown in Fig. 3(a), when we pass the minimum point a, a new component appears

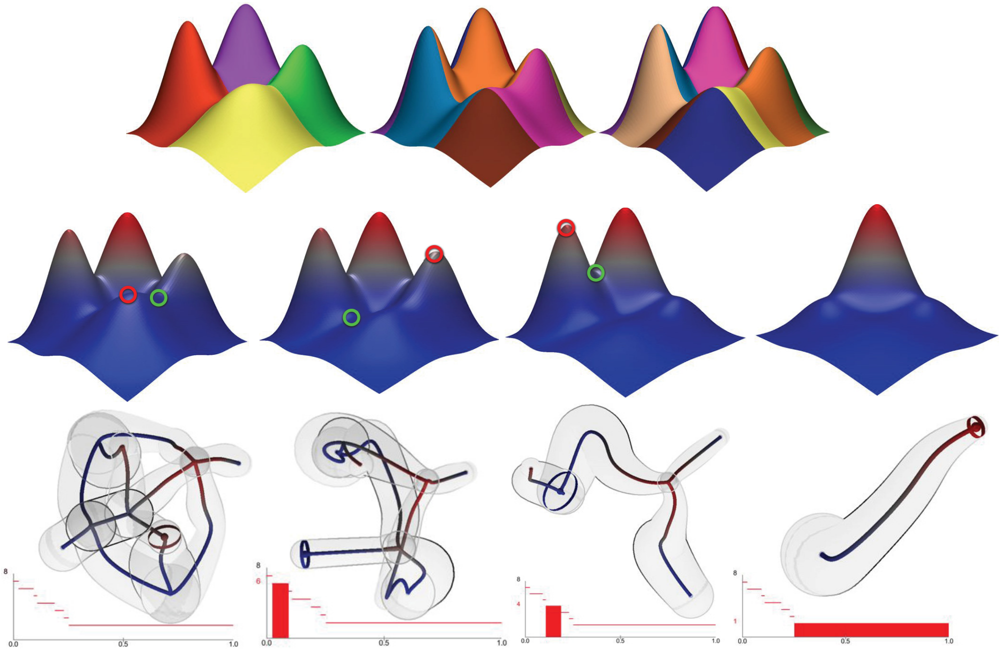


Figure 4: An illustrative example based on a 2D height function. Top: descending manifolds, ascending manifolds and Morse-Smale complex. Middle: progressive simplification of its critical points (maxima are red, saddles are green). Bottom: topological summaries shown at various scales (see Section 3).

in the sublevel sets, which is represented by the minimum a, with a *birth* time $f(a)$. Similarly when we pass the minimum b and c, two new components are born with birth time $f(b)$ and $f(c)$, respectively. When we pass the maximum d, two components represented by a and c are merged and the maxima is paired with the younger (higher) of the two minimum that represent the two components, that is, d and c are paired, where $f(d)$ is the *death* time of the component represented by c. We define the *persistence* of the pair to be $f(d) - f(c)$, which corresponds to the significant of a topological feature. We then encode persistence in the *persistence diagram* of f, $\mathrm{Dgm}(f)$, by mapping the critical point pair to a point $(f(c), f(d))$ on the 2D plane. Similarly we pair e with b and f with a, resulting two more points in $\mathrm{Dgm}(f)$. For technical reasons, the diagonal is considered as part of the persistence diagram that contains an infinite number of points.

Recent results show that persistence diagrams are stable under small perturbations of the functions [13, 14]. Let $p = (p_1, p_2), q = (q_1, q_2)$ be two points in the persistence diagram, let $||p - q||_\infty = \max\{|p_1 - q_1|, |p_2 - q_2|\}$. For functions $f, g : \mathbb{R} \to \mathbb{R}$, $||f - g||_\infty = \sup_x |f(x) - g(x)|$. The *bottleneck distance* between two multi-sets of points in $\mathrm{Dgm}(f)$ and $\mathrm{Dgm}(g)$ is

$$d_B(\mathrm{Dgm}(f), \mathrm{Dgm}(g)) = \inf_\gamma \sup_x ||x - \gamma(x)||_\infty,$$

where $x \in \mathrm{Dgm}(f)$ and $y \in \mathrm{Dgm}(g)$ range over all points, and γ ranges over all bijections from $\mathrm{Dgm}(f)$ to $\mathrm{Dgm}(g)$ [14]. The Stability Theorem states that the persistence diagrams satisfy: $d_B(\mathrm{Dgm}(f), \mathrm{Dgm}(g)) \leq ||f - g||_\infty$. This is illustrated in Fig. 3(b).

A Complete Illustrative Example. Finally for a complete intuitive understanding of the above concepts, we give the following example of another 2D height function, as shown in Figure 4.

The top of the figure illustrates its corresponding descending manifolds, ascending manifolds and Morse-Smale complex, while the middle of the figure showcases the progressive simplification of its critical points: at the finest level there are four maxima in the data, but as we increase the scale, neighboring topological features are merged where the ascending gradient flow is directed to its more salient neighboring maxima. Circled points correspond to the critical points that are removed at each simplification step. We thus build a filtration of segmentations where Morse-Smale crystals are merged based on the persistence value of their associated extrema.

3 High-dimensional Data Analysis and Visualization

3.1 Motivation

Dynamic Probabilistic Risk Assessment (PRA) methodologies [19] couple numerical simulation tools and time-dependent stochastic models (i.e., probabilistic failure models or parameter uncertainties), to perform system safety analysis. Widely used dynamic PRA methodologies are based on Monte-Carlo [45] or Dynamic Event Tree algorithms [26]. The common underlying idea is to run a large number of simulations (by employing system simulators) where values of system stochastic parameters (e.g., timing of failure of a specific component or an uncertain parameter) are sampled from their own distribution at each run. This type of PRA analysis can be very time-consuming when a large number of stochastic parameters are considered and when large and complex system simulators are used. Moreover, a large volume of data is typically generated. Such large amounts of information can be difficult to organize for extracting useful information. Furthermore, it is often not sufficient to merely calculate a quantitative value for the risk and its associated uncertainties. The development of risk insights that can increase system safety and improve system performance requires the interpretation of scenario evolutions and the principal characteristics of the events that contribute to the risk.

The need for software tools able to both analyze and visualize large amount of data generated by Dynamic PRA methodologies has been emerging only in recent years. A first step has been shown in [52,53] using clustering-based algorithms which focus more on the analysis part than the visualization side.

In this section, we present a software tool that provides scientists and domain experts with an interactive analysis and visualization environment for understanding the structures of high-dimensional nuclear simulation data. Our tool adapts and extends the innovative framework called HDViz first proposed by Gerber et. al. [29] in exploring high-dimensional scalar functions, and applies the underlying techniques to nuclear reactor safety analysis and visualization. Our tool includes a host of various analysis and visualization capabilities. We describe each of these capabilities on a modular basis, by explaining the underpinning theories and presenting usage cases. The software segments the domain of a high-dimension function into regions of uniform gradient flow by decomposing the data based on its approximate Morse-Smale complex. Points belonging to a particular segment have similar geometric and topological properties, and from these we can create compact statistical summaries of each segment. Such summaries are then presented to the user in an intuitive manner that highlights features of the dataset which are otherwise hidden in a global view of the data. In addition, the visual interfaces provided by the system are highly interactive and tightly integrated, providing users with the ability to explore various aspects of the datasets for both analysis and visualization purposes.

3.2 A Note on Data Standardization

The high-dimensional data may be pre-processed with a standardization process. Since different parameters may be measured on different scales and the range of values differ from each dimension, some parameters may dominate the results of the analysis. Various methods exist for data standardization [31, 41, 56]. Several methods under considerations are:

- **Z-score scaling**: values V of each dimension are recomputed as $V - \text{mean}(V)/\text{std}(V)$; therefore all input parameters have the same means (0) and standard deviation (1) but different ranges.
- **Robust Z-score scaling**: if values of each dimension are expected to have outliers, a more robust scaling involves substituting the outlier-insensitive median and median absolute deviation (MAD) for mean and standard deviation in the Z-scoring scaling, that is, V is recomputed as $V - \text{median}(V)/\text{MAD}(V)$, where the MAD of a set of variables $\{x_i\}_{i=1}^n$ is defined as $\text{median}\left(\{|x_i - \text{median}\left(\{x_j\}_{j=1}^n\right)|\}_{i=1}^n\right)$.
- **Dividing by the standard deviation**: values V of each dimensions are recomputed as $V/\text{std}(V)$; then all input parameters share the same variances 1 but have different means and ranges.
- $[0,1]$**-Scaling**: V is recomputed as $V - \min(V)/(\max V - \min V)$. The input variables have the same ranges but different means and standard deviations.
- **Dividing by the range**: V is recomputed as $V/(\max V - \min V)$. The input variables may have different means, variances and ranges; however the ranges are likely comparable.

All these above standardization techniques may lead to different topological summaries. The appropriate choice of method depends on the nature of the dataset and the convention in the field. Sometimes it is possible that the data may not need to be standardized. When such a choice is not obvious, we provide the end users multiple choices and offer the corresponding analysis and visualization results.

3.3 Analysis and Visualization Modules

In this section, we describe each analysis and visualization module within our integrated system that are either part of the original capabilities provided by HDViz [29] or part of our extension.

3.3.1 A 6D VR$_2^+$ Case Study

We demonstrate our infrastructure with a 6D example dataset from nuclear plant safety analysis. The data is extracted from a VR$_2^+$ nuclear reactor simulator and represents an ensemble of 10000 simulation trials where a SCRAM is simulated due to a failure in the system. A SCRAM event is when the control rods of the reactor are inserted into the core in order to prevent overheating of the reactor core. The output variable is the peak coolant temperature (PCT[1]), measured in Kelvin. The domain scientists are interested in what combination of conditions (in the form of input parameters) can cause potential reactor failure (i.e. nuclear meltdown witnessed by PCT exceeding a threshold value). The input space is defined by six parameters:

- **PumpTripPre** - the minimum pressure (MPa) in the heat exchange pump causing the SCRAM to trip
- **PumpStopTime** - the relaxation time (sec) of pump's phase-out
- **PumpPow** - end power of the pump
- **SCRAMtemp** - the maximum temperature in the system causing the SCRAM to trip
- **CRinject** - the control rod position at the end of SCRAM
- **CRtime** - the relaxation time (sec) of the control rod system.

The visualization is based on the original parameter values, not after data standardization.

3.3.2 Topological Summary

The visual interface designed for topological summary is inherited and extended from the capabilities provided by HDViz [29] with additional capacity for user interactivity. Considered as the main visual display of our software, this interface summarizes each Morse-Smale crystal into a 1D curve in high-dimensional space which is then projected onto a viewable 3D space. The interface encodes three steps (detailed in [29], with a high level description below), to arrive at a 3D representation for analysis and visualization of the d-dimensional scalar function f, defined on a set of sampled points $\mathbb{X}$. (1) *Morse-Smale approximation*: We approximate the Morse-Smale crystals, $\mathbb{X}_i$ and $\mathbb{Y}_i = f(\mathbb{X}_i)$, in high dimension using an approximate k-nearest neighbor graph [3], as described in Section 2. (2) *Geometric summaries:* For each crystal of the Morse-Smale complex, since each point has similar approximated monotonic gradient behavior, a geometric summary is constructed by an inverse regression, yielding a 1D curve r_i in the d-dimensional domain of f. Intuitively, $r_i(y)$ (where $y \in \mathbb{Y}_i$) yields a representation of the crystal as the average of the function values of level sets within the crystal, see [29] for its detailed derivation. (3) *Dimension reduction:* The set of regression curves can be represented by a graph embedded in $\mathbb{R}^d$ with each edge corresponding to a curve and vertices corresponding to extremal points. For visualization, we embed this graph into 2D preserving the spatial relation among extrema and the geometry of the crystals that connect them. First, vertices are embedded using PCA or ISOMAP [64]. Second, edges are embedded individually through their first two principle components. And third, the resulting 2D curves are attached to the projected vertices through affine transformations. The third dimension is reserved for the output parameter.

Visual Components. The visual components of the above framework are shown in Figure 5 for a simple 2D function with one crystal (example revisited from Figure 4). Each curve is encased

[1] Note that PCT used here does not stand for peak clad temperature.

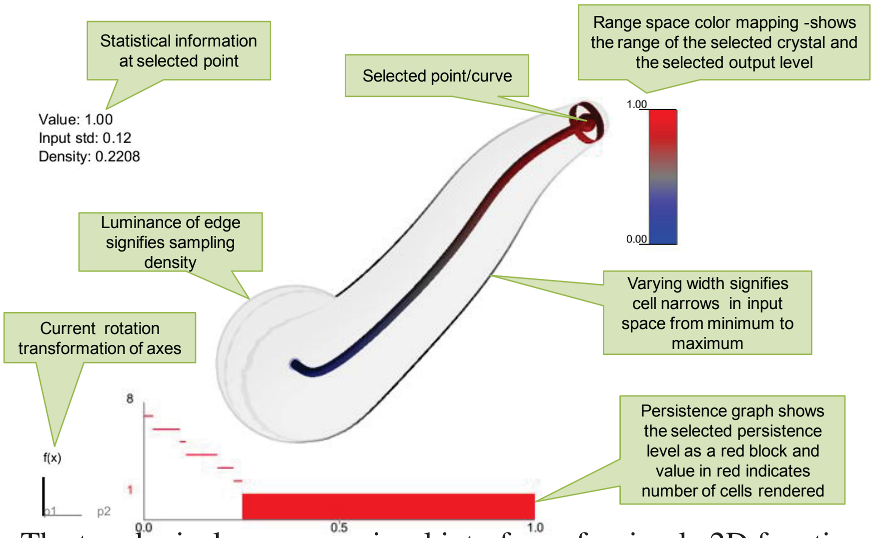

Statistical information at selected point
Value: 1.00
Input std: 0.12
Density: 0.2208
Selected point/curve
Range space color mapping -shows the range of the selected crystal and the selected output level
1.00
0.00
Luminance of edge signifies sampling density
Varying width signifies cell narrows in input space from minimum to maximum
Current rotation transformation of axes
f(x)
p1
p2
8
1
0.0
0.5
1.0
Persistence graph shows the selected persistence level as a red block and value in red indicates number of cells rendered

Figure 5: The topological summary visual interface of a simple 2D function.

in a transparent tube where the width of the tube represents the “spread” of the data at a particular scale, and the luminance of the tube encodes the density of data points within each crystal. Users are given the flexibility to view the topological summary of a high-dimensional function by switching between PCA and ISOMAP projections, and using affine transformations to manipulate the projection directly on the screen. In order to preserve the “width” of a crystal at a given scale, we compute the standard deviation with respect to each input parameter and also a single *average* standard deviation across all input parameters. The latter is direction-independent and can therefore be used as a generalized width of the data at a particular output level. The radius of the outer transparent tubes are defined by this direction-independent standard deviation. The last visual cue is the darkness of the edge of the transparent tube. Where the sampling density is high, the outline of the tube is drawn black and as the sampling density decreases, the luminance of this edge increases. To enable multi-scale analysis, we use a modified version of the persistence diagram [24], referred to as the *persistence graph*. This is shown as a visual component at the bottom of Figure 5. It shows the number of Morse-Smale crystals (y-axis) as a function of scale (i.e. x-axis, persistence threshold normalized by the range of the dataset). A selected scale is drawn with a red box and a corresponding number of Morse-Smale crystals is displayed along the y-axis in red. Stable features are considered as those that exist over a large range of scales (i.e. a sequence of persistence simplification with increasing scales), which correspond to long horizontal lines in the persistence graph.

We demonstrate how the topological summary visual interface allows the users to explore the data at multiple scales. We revisit our simple dataset under several levels of persistence simplification in Figure 4 bottom. Here the leftmost image shows the full resolution of data with four local maxima, while each subsequent image reduces the number of maxima by one until we are left with a single crystal describing the gradient flow from the global minimum to the global maximum. The numbers in red, shown in the persistence graphs, indicate the total number of crystals displayed, from left to right, as 8, 6, 4 and 1. Instead of giving the users a representation of the data at a fixed scale, we provide an interactive platform to help them differentiate features from noise through multi-scale analysis and to choose the appropriate scale based on domain knowledge and data characterization.

A Simple Example of Geometric Summary. Figure 6 shows several examples of simple 2D surfaces represented using the techniques described in this section. Each function has one local maximum and a single local minimum surrounded by a flat area. Using an evenly sampled grid, the density of points available at each level is mapped to the color on the edge of the transparent tubes. The widths of the tubes vary with respect to the spread of the data at each level set. Note how the sharp spike in (a) is represented in the topological display. The steep peak covers a small area and the small width of the transparent tube demonstrates this in the area near the maximum. (b) has a wider area surrounding the peak and while its representing curve remains a single straight line, the tube surrounding it changes behavior to account for the width of each level set. The plateau function in (c) even maps the discontinuity in the curve by making a sudden jump from a low value to a high value. Note how the edge of the tube has high luminance in the middle section denoting a lack of data used to compute this section, whereas the most densely sampled region is at the maximum value which has a black outline.

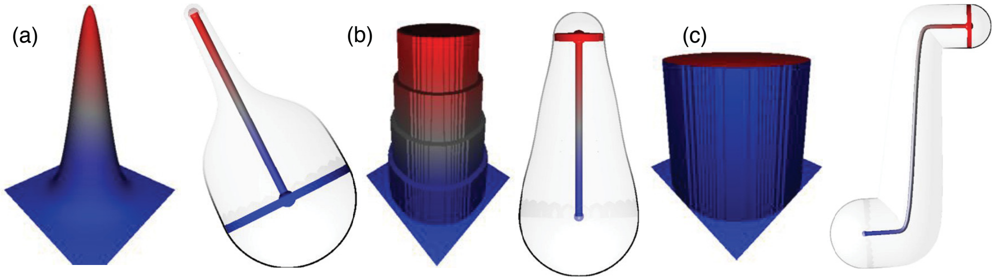


Figure 6: Three simple surfaces represented using geometric summary tubes.

6D Demo Dataset. As a final example, we illustrate our interface with our 6D demo example in Figure 7 under multiple scales. This dataset contains 753 individual crystals at the finest level, though due to low persistence and visual clutter caused by these crystals, we restrict our analysis to only a handful of the most salient crystals. During the persistence simplification steps, the topological summaries consist of 6, 3 and 1 Morse-Smale crystal(s), respectively, based on their long horizontal lines in the persistence graph (that likely correspond to stable features). At each of the shown persistence levels, the dataset is characterized by a single global minimum with high sampling density. The widths of all the crystals at the minimum is expansive compared to the widths at the maxima. We could infer from this analysis result that most of the data points (simulations) represent lower PCTs and the conditions to reach these lower PCTs varies widely in the domain space. On the other hand, the maximum PCTs are reached at more specific input ranges which is made clear through the narrow widths of the tubes (of one standard deviation) surrounding the red maxima. For these chosen scales, further explorations could be done to understand the different combinations and correlations among input parameters that lead to 6, 3 and 1 local maxima of PCT, respectively.

3.3.3 Statistical Summary

The visual interface shown in Figure 8 demonstrates statistical and geometric (i.e. gradient) information associated with the selected point in the topological summary interface. Each input

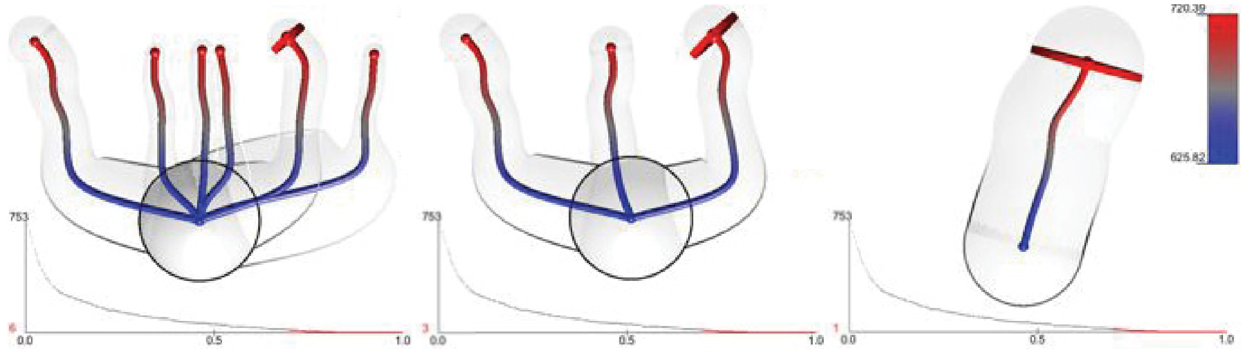


Figure 7: 6D demo dataset: Topological summaries shown at various scales.

dimension, or coordinate, is viewed as an inverse function of the output parameter. In the left column of the statistical summary window, each horizontal axis describes the range of values of each input parameter and the coordinate mean and coordinate standard deviation associated with the selected point. The right column encodes the gradient information, that is, the change in the output with respect to the change in each input parameter. In Figure 8, we can see that three parameters have quite large standard deviations, PumpStopTime, SCRAMtemp, and CRtime, whereas the other parameters vary much less. This indicates a certain characteristic associated with the chosen crystal with respect to these three parameters.

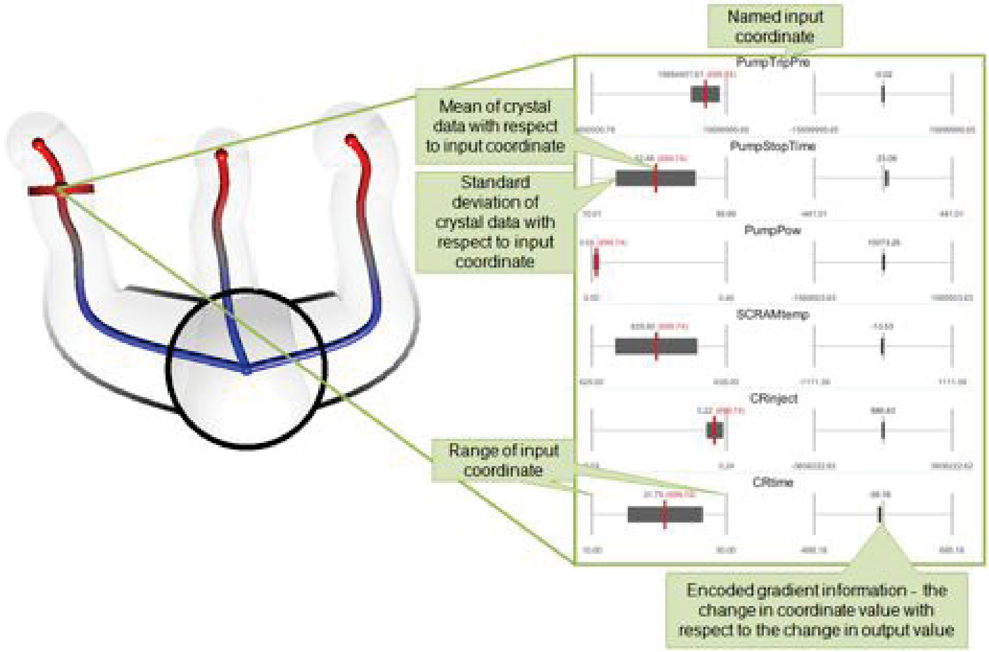

Named input coordinate
PumpTripPre
Mean of crystal data with respect to input coordinate
PumpStopTime
Standard deviation of crystal data with respect to input coordinate
PumpPow
SCRAMtemp
CRinject
Range of input coordinate
CRtime
Encoded gradient information – the change in coordinate value with respect to the change in output value

Figure 8: 6D demo dataset: A snapshot of the statistical summary with highlighted visual components.

3.3.4 Inverse Coordinate Plots

In the inverse coordinate plots, each input parameter is considered as a 1D function of the output variable. This visual interface is shown in Figure 9 for our 6D example. On the left, at the persistence level with three crystals, the interface displays the inverse coordinate plot for data points associated with the selected Morse-Smale crystal in the topological summary interface. On the right, at the persistence level with six crystals, the interface shows combined inverse coordinate plots associated with all the crystals that share the same local minimum. For the selected crystal(s), the regression curve is drawn in the inverse coordinate plots with a grey tube representing the parameter-specific standard deviation.

From the right image in Figure 9, we see the top set of axes where each of the six regression curves is readily distinguishable with respect to different ranges of values for PumpTripPre, whereas in the lower plots they vary little toward the left and only slightly at the right. On the other hand, parameters such as PumpPow and CRinject result in high temperatures only at specific levels. This conclusion is based upon the tight configuration of points resulting in high PCT and it is also supported by the consistent locations of the mean values across all crystals and the low standard deviations of these parameters.

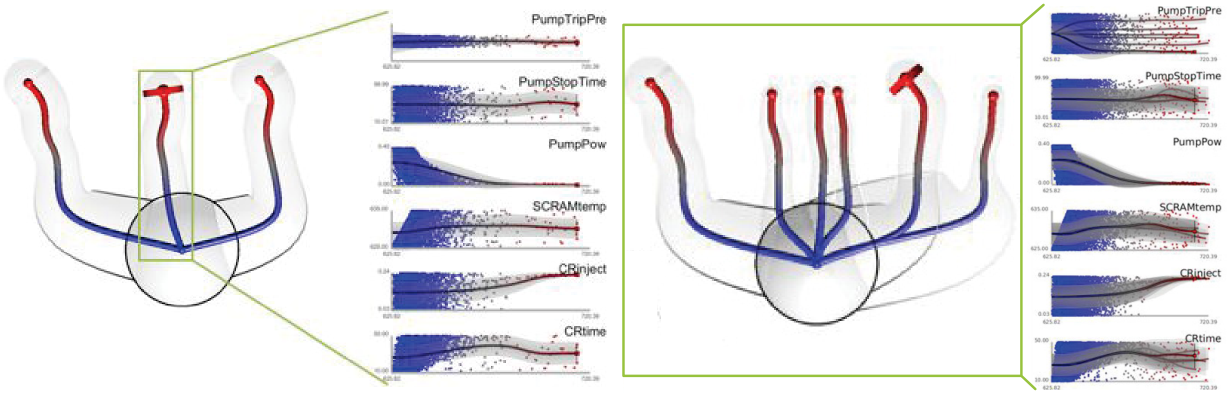


Figure 9: 6D demo dataset: Inverse coordinate plots shown for the highlighted crystal for the three crystal (left) and six crystal (right) case.

3.3.5 Interactive Projection, Parallel Coordinate Plots and Pairwise Scatter Plots

As the number of dimensions increases for a high-dimensional function defined on the sampled point set, selecting the interesting projection dimensions while interacting with the high-dimensional function could become counter-intuitive. We design an interactive visualization interface that provides simple and fully explanatory pictures that give comprehensive insights into the global structure of the high-dimensional function. Based on hypervolume visualization techniques developed in [4], our basic idea is a generalization of direct parallel projection methods. We first create an independent viewing system that scales with the number of dimensions where the user is allowed to manipulate how each axis is projected. We then apply these manipulations to project the geometric summaries from the functional space to the screen space [4]. Users are provided with a wheel of labeled axes which they can manipulate by stretching, contracting, and rotating. The geometry of the inverse regression curves therefore is meaningfully preserved. With a single manipulated projection displayed at a time, the interactive nature of such a tool provides the user with intuition that is otherwise lost in a PCA or ISOMAP projection. An example for the 6D demo dataset is shown in Figure 10. Currently, this system only supports a view of the geometric summaries, but a possible extension is to create a full hypervolume visualization of the raw high-dimensional data points as proposed in [4].

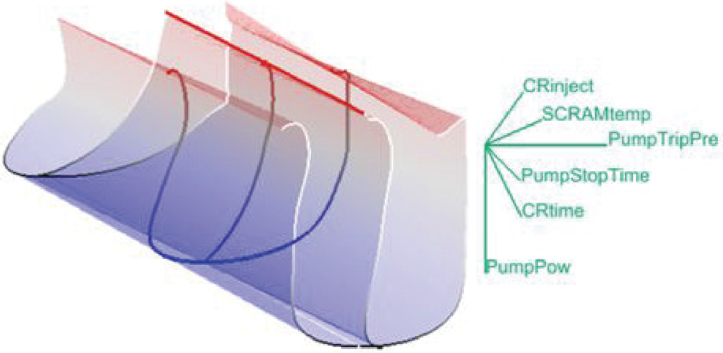


Figure 10: 6D demo dataset: Interactive Projection Visual Interface for the three crystal case. Left: projection of the high-dimensional inverse regression curves and their associated standard deviation tubes, based on the mapping shown in the right. Right: each input dimension is mapped to a single green line segment, which the user can manipulate by stretching and rotation, to emphasize or diminish the effect of a particular dimension on the projection.

Furthermore, we also use parallel coordinate [39] plots (Figure 11 left) where each input parameter is represented as a vertical axis, and a single line connecting parallel axes represents the adjacent dimensions of a given data point. In addition, we present a matrix of scatter plots (Figure 11 right) where each pair of input parameters are plotted as the *x* and *y* axes with the output variable mapped by color. We can see in both plots, the selected crystal of our 6D demo dataset has the defining characteristic of only extracting a portion of the domain values for the PumpTripPre parameter. Such features in the visualization can help us infer correlations among input parameters, and their influence on the output parameter.

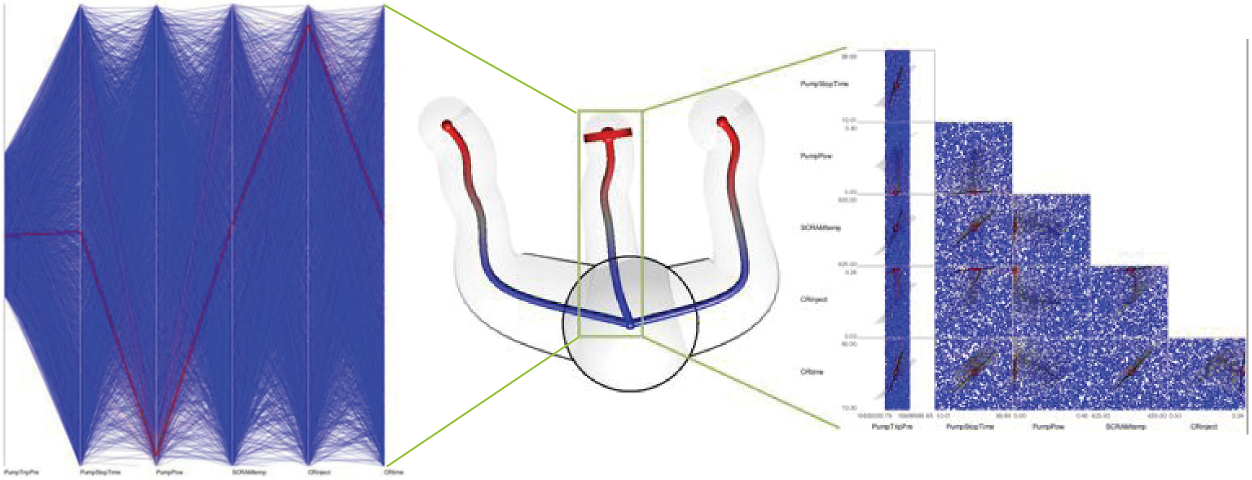


Figure 11: 6D demo dataset, with parallel coordinate plots (left) and pairwise scatter plots (right) for a selected crystal.

3.4 A 21D VR_2^+ Case Study

We perform a similar analysis on a more complicated simulation involving 21 input parameters where the output variable is again peak coolant temperature. As stated previously, the dataset consists of 10,000 individual simulations where the parameters include the parameters of the 6-dimension dataset in addition to 15 others. As shown in the persistence graph of the topological summary visual interface (Figure 12), the dataset at its finest scale consists of a large number of unstable features (732 Morse-Smale crystals in total). We then simplify such a complex structure by increasing the persistence threshold and arrive at a relatively stable structure with five Morse-Smale crystals.

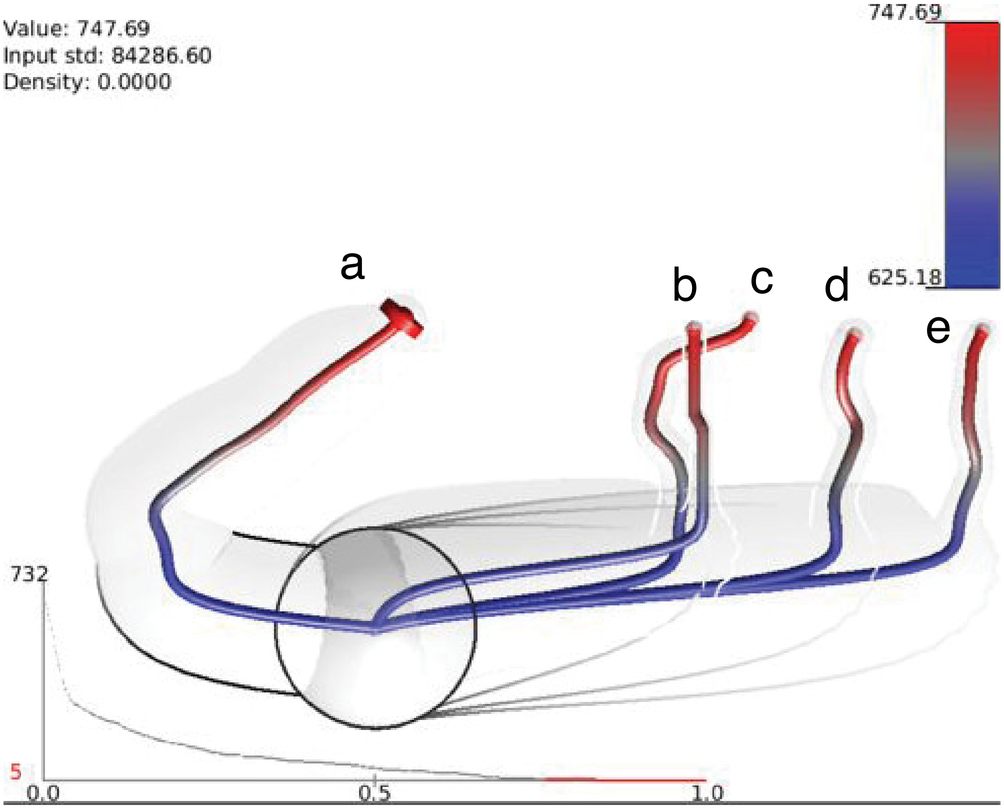


Figure 12: The topological summary interface for the VR_2^+ 21D dataset: at a certain scale, there are five Morse-Smale crystals sharing one global minimum.

Based on its topological summary (Figure 12), it is clear that the overall simplified structure of the simulation space is similar to that of the 6D dataset. That is, all Morse-Smale crystals share a single global minimum and the data points that reside at or near such a minimum constitute a large percentage of the total sampled points. The tubes near each maxima are relatively narrow, representing a small number of input parameter settings that result in the production of high PCT values.

Selecting a specific crystal in the topological summary shows us where in the range space the crystal is most heavily sampled, but this interface tells us little about how much data is associated

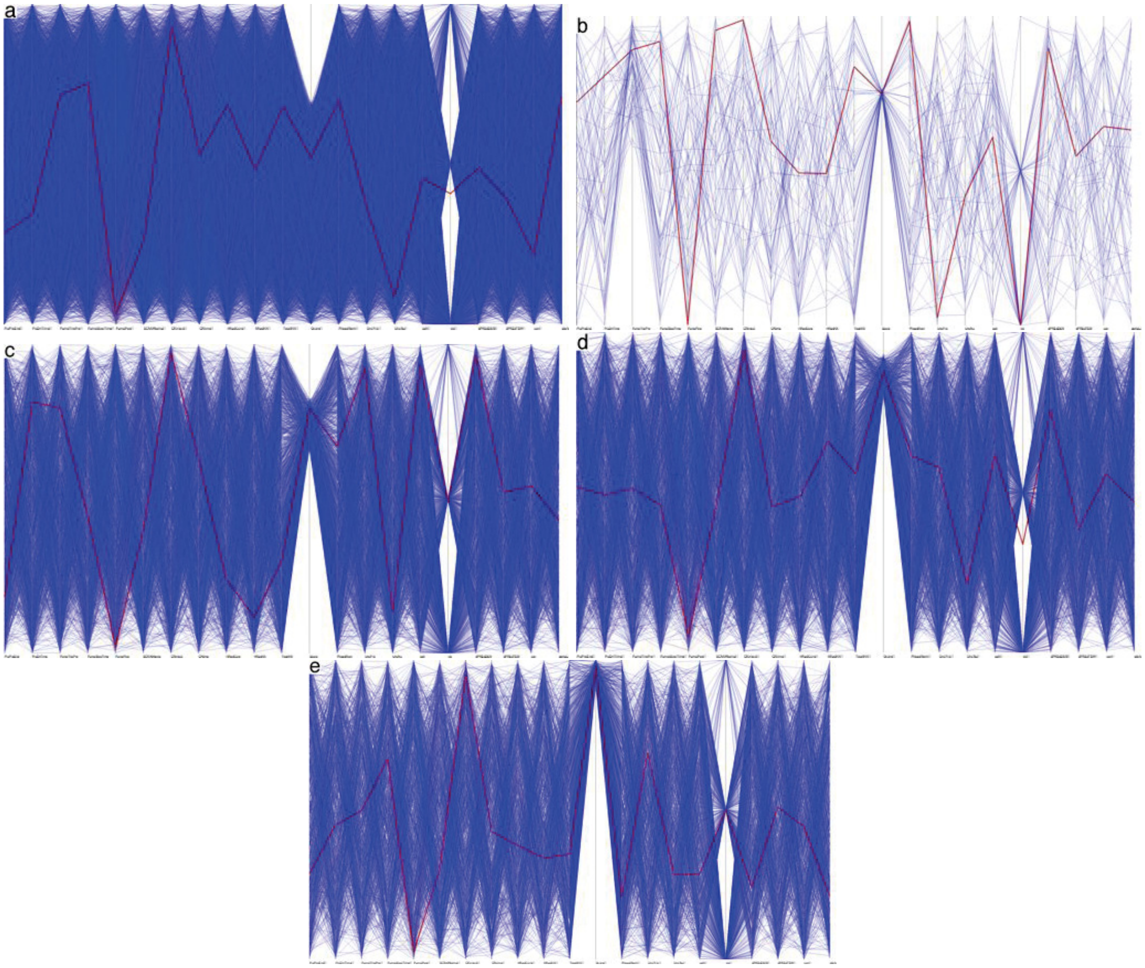


Figure 13: The parallel coordinate plots of the VR_2^+ 21D dataset. They are ordered from left to right, from top to bottom, for crystals *a* through *e*.

with an entire crystal. On the other hand, the parallel coordinate plots (Figure 13) readily demonstrate that the crystal *b* contains much fewer data points than the other crystals. Furthermore, we note that each crystal targets separate ranges of values for the Qcore parameter (12-th axe from the left) which is not clearly evident in the other visual interfaces.

In addition, again we could also combine the inverse coordinate plots of all Morse-Smale crystals together into one stable manifold and visualize the combined plot in Figure 14. The visual interface shows that PumpPow and CRinject remain the dominating factors in terms of reaching a high PCT, that is, only a high CRinject value and a low PumpPow value will yield maximum PCTs for all crystals.

Finally, we can also construct the pairwise scatter plots (Figure 15). As discussed before, as the dimension of the dataset increases, such plots eventually become a dense visual cluster, causing the interface to be ineffective. We are investigating other ways to present these pairwise relations among input parameters, such as focused or sub-sampled views.

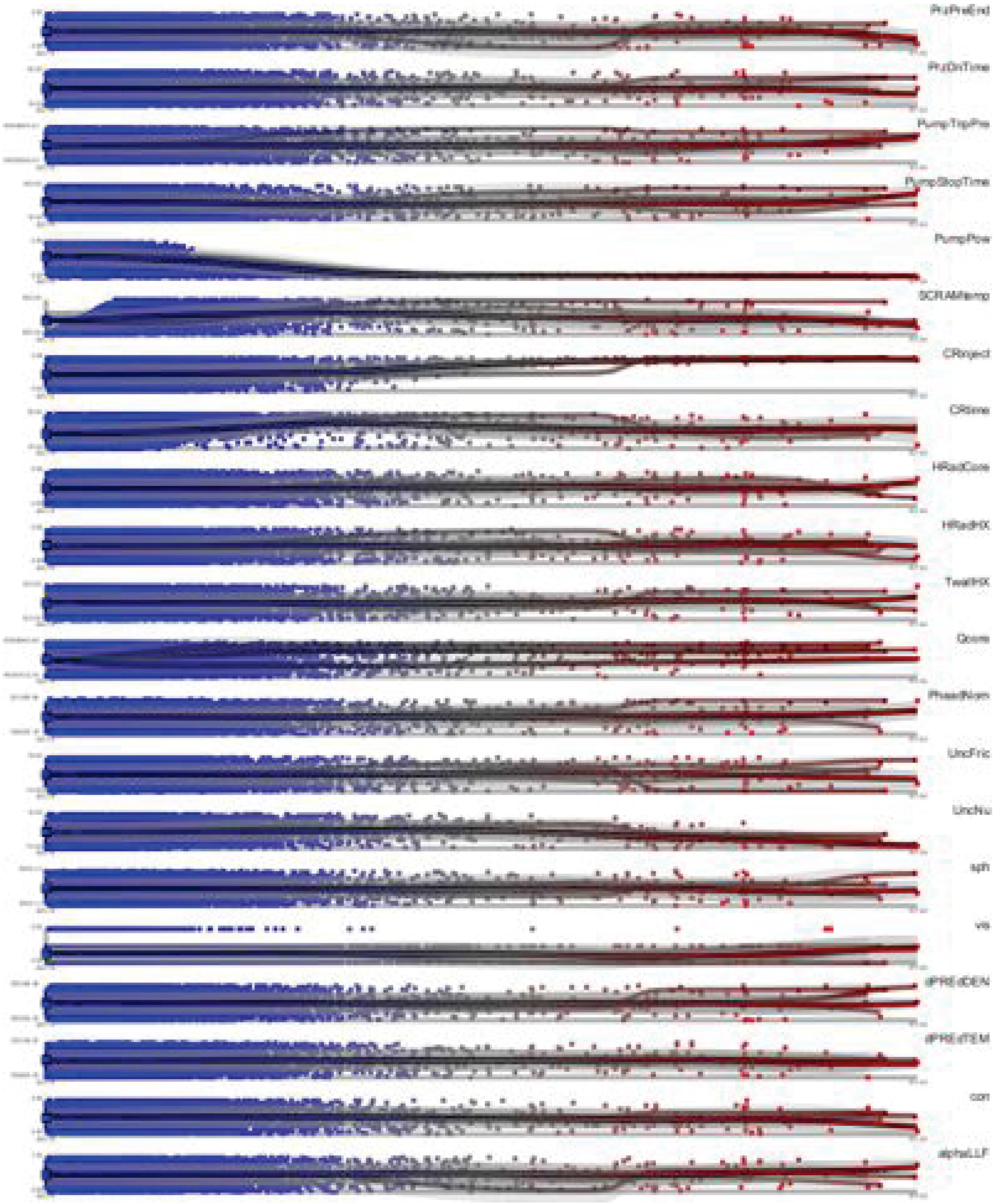


Figure 14: VR_2^+ 21D dataset: Inverse coordinates plots for all Morse-Smale crystals combined (into one stable manifold).

3.5 Analysis and Visualization with Z-score Scaling Data Standardization

We first revisit both the 6D and 21D case studies with data standardization based on the Z-score scaling. Here we use a different coloring scheme that emphasizes clustering structure instead of function value. The shorthand notion we adapt for each case encodes the dimension, output variable of interest, scenarios included and number of crystals obtained. For example, case **6D-PCT-All-5C** means a 6D case analysis for all the scenarios, where the output variable is PCT, and the obtained results include 5 crystals.

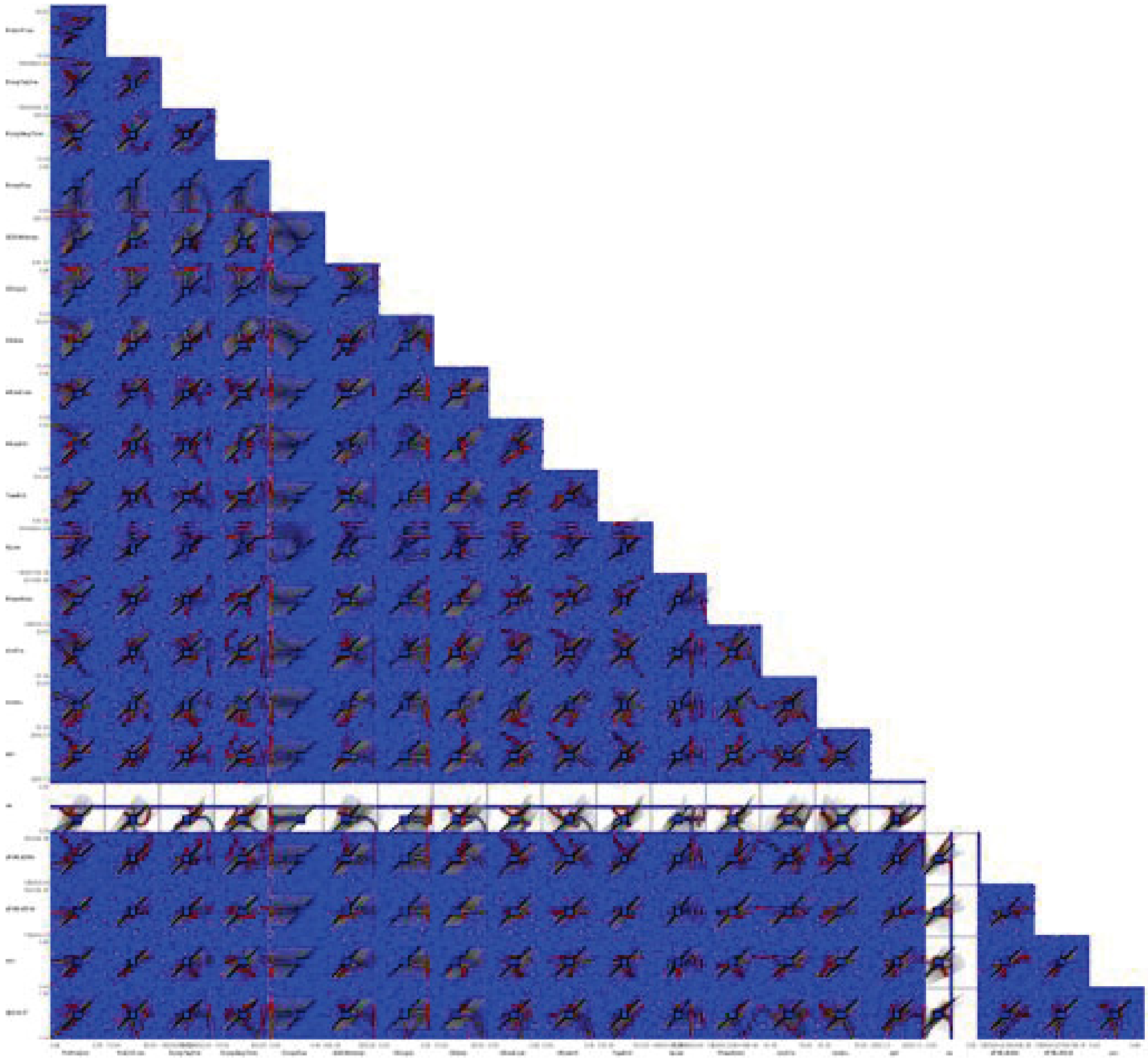


Figure 15: The pairwise scatter plots of the VR_2^+ 21D dataset.

6D-PCT-All-5C. We analyze this 6D standardized dataset in the setting where there is one global minimum with a value of 625.82 shared among five separate crystals, each with a distinct maximum. The maximum values in increasing order are: 711.08 (magenta), 714.11(red), 715.00(orange), 718.69(blue), and 720.39 (green). The most compelling evidence to note is that a low PumpPow and a high CRinject are necessary, but not sufficient conditions to reach a high PCT value. The remaining 4 dimensions distinguish the 5 crystals within the input domain near each of their distinct maxima located on the righthand side of the coordinate plots in Figure 16.

21D-PCT-All-4C. From Figure 19, we note there is a stable topological decomposition that consists of four separate crystals all sharing a global minimum point with a peak coolant temperature value of 625.18. The output values of the local maximum for each crystal are, in increasing order: 742.36 (blue), 743.15 (magenta), 747.24 (red), and 747.69 (green).

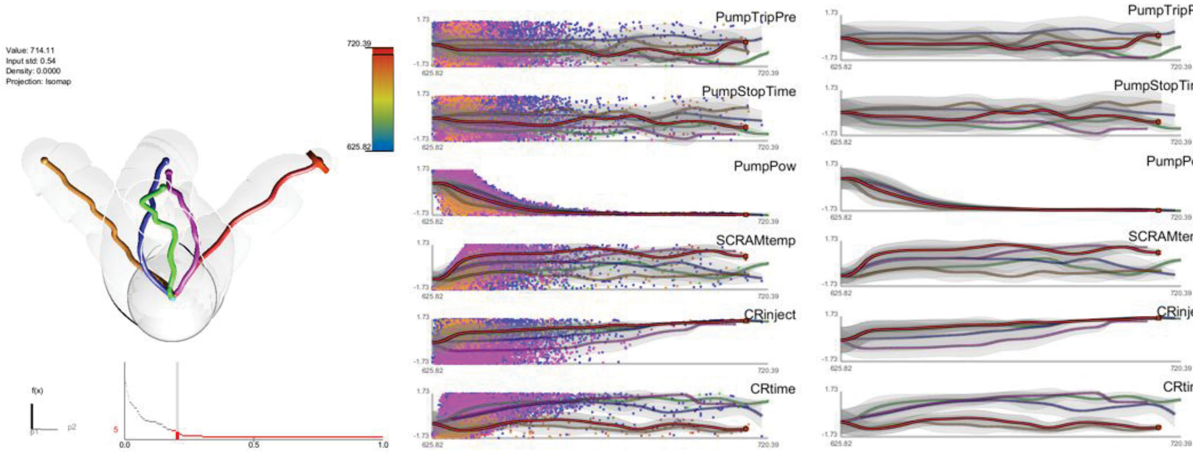


Figure 16: **6D-PCT-All-5C**: topological summary (left), inverse coordinate plots with (middle) and without (right) points projection. Points and regression curves are colored by cluster (crystal) memberships.

As in the 6D case, we see that low PumpPow and high CRinject are necessary conditions to reach high PCT values. After closer analysis of several input dimensions, we can note that other parameters like CRtime, PumpTripPre, PumpStopTime, UncNu, and QCore do not vary when comparing the spread of data points and the regression curve of the individual coordinate plots of Figure 19 (top and right image). The correlation of these parameters to the output is less strong than the CRInject and PumpPow since the data is spread farther in each dimension and the gray tubes are not as tightly bound to the regression curves in these dimensions. In the bottom row of images, we highlight dimensions that showcase significant features of each crystal. For example, in the bottom left image, we see that the dimensions of SCRAMtemp and vis have high and low settings, respectively, that lead to the local maximum of the magenta crystal, whereas a low PHeadNom and dPREdDEN setting result in the local maximum of the blue crystal. The two higher maxima achieved by the red and green crystals are harder to distinguish from all of the other three crystals, but we can differentiate them from each other by looking at the dimensions highlighted in the lower left image of Figure 19. Namely, these are alphaLLF, con, PrzDnTime, and TWallHX.

3.6 Analysis and Visualization with $[0, 1]$-Scaling Data Standardization

For completeness, we then present some visualization results under $[0, 1]$-scaling of the data. Here we use a different coloring scheme that emphasizes clustering structure instead of function value.

6D-PCT-All-3C. The result is shown in Figure 18. Figure 18 shows a stable setting in the topological hierarchy where there is a single global minimum with a peak coolant temperature of 625.82. There are three local maxima. One of the maxima, identified as the maximum of the green crystal, has a significantly lower value (695.64) than the blue (718.69) and red (720.39) crystals. As in the other analysis on this same dataset, we see that a low PumpPower and a high CRinject value are necessary conditions for a resulting high peak coolant temperature. PumpStopTime and

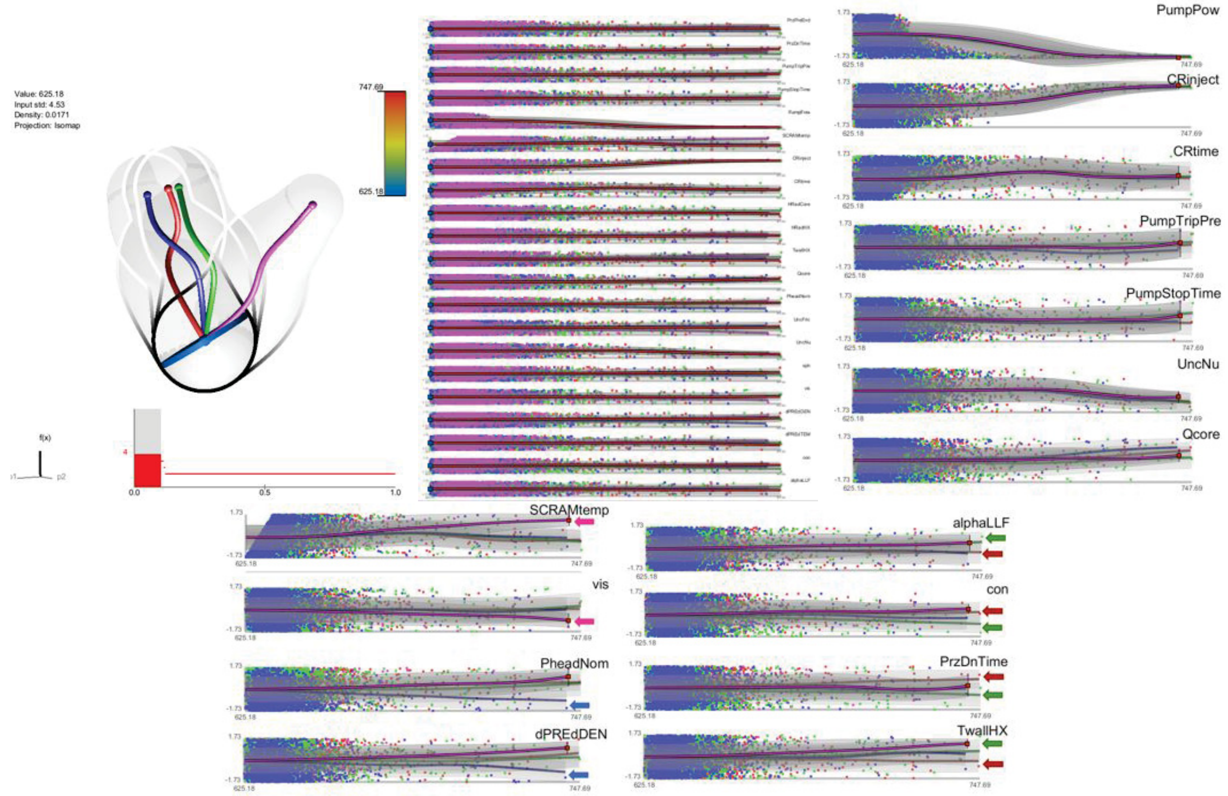


Figure 17: **21D-PCT-All-4C**: topological summary (left), inverse coordinate plots with (middle) and without (right) points projection. Points and regression curves are colored by cluster (crystal) memberships.

SCRAMtemp are the most differentiating dimensions for separating the three clusters when looking at the inverse coordinate plots. To a lesser extent, we can distinguish the three crystals in the CRTime dimension. And in the PumpTripPre plot, the blue and green crystals shared a similar behavior which is distinguishable from the regression curve of the red crystal.

21D-PCT-All-4C. In this 21D case where the output variable is peak coolant temperature (PCT), we analyze the dataset at a setting with four clusters in the topological segmentation. At this level, the data has a single global minimum shared by all four crystals that has a value of 625.18. The blue crystal has the lowest maximum value of 742.46, followed by the magenta crystal with a value of 743.15, then the red crystal with a value of 747.24 and finally the green crystal barely edges out the red with a maximum value of 747.69. As in the 6D case, we see that where there is a high PCT, the CRinject value will be high and the PumpPower will be low. Other parameters that do not vary between the separate maxima, as evidenced by the lack of separation of the regression curves toward the right end of the scatter plots in Figure 19, include: PumpTripPre, CRTime, HRadHX, Qcore, and UncNu. For distinguishing the magenta crystal from the others, we would note its distinctively high value of SCRAMtemp at its maximum, or its low value of UncFric or vis. Similarly, the blue crystal has its own parameters that differentiate it from the other three crystals, namely low values for both PheadNom and dPREdDEN. All that remains is to separate the red

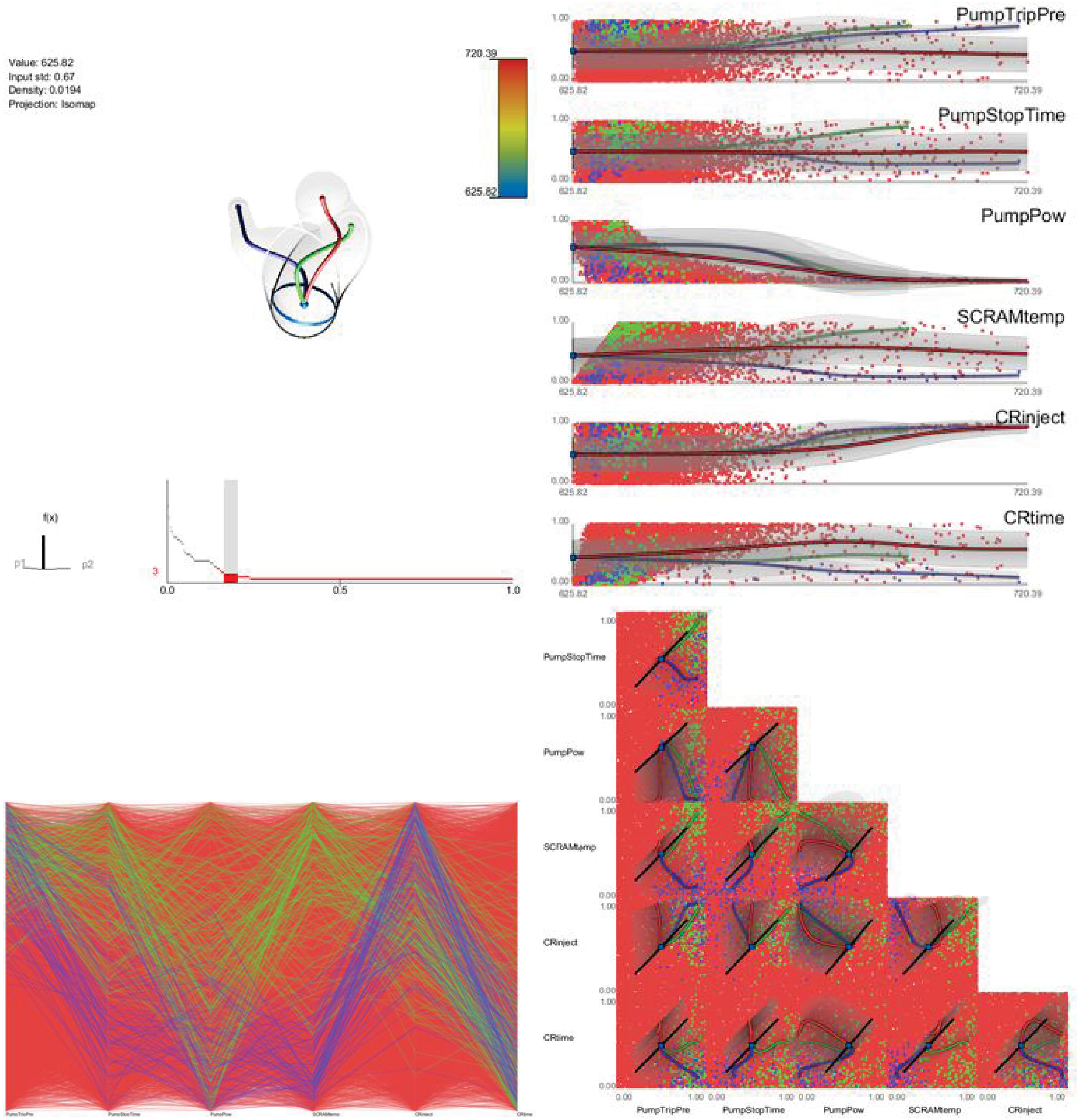


Figure 18: **6D-PCT-All-3C**: Clockwise from top left: Topological skeleton, inverse coordinate plots for each dimension, pairwise scatter plots showing data projected onto the plane spanned by every pair of input dimensions, and the parallel coordinate plot with every simulation colored by cluster id.

and green crystals from one another, since we can already distinguish them from the blue and magenta. For this task, we look at PrzDnTime, HRadCore, TwallHX, con, and alphaLLF which are highlighted in Figure 19 showing each of the red and green crystal's particular setting with colored arrows.

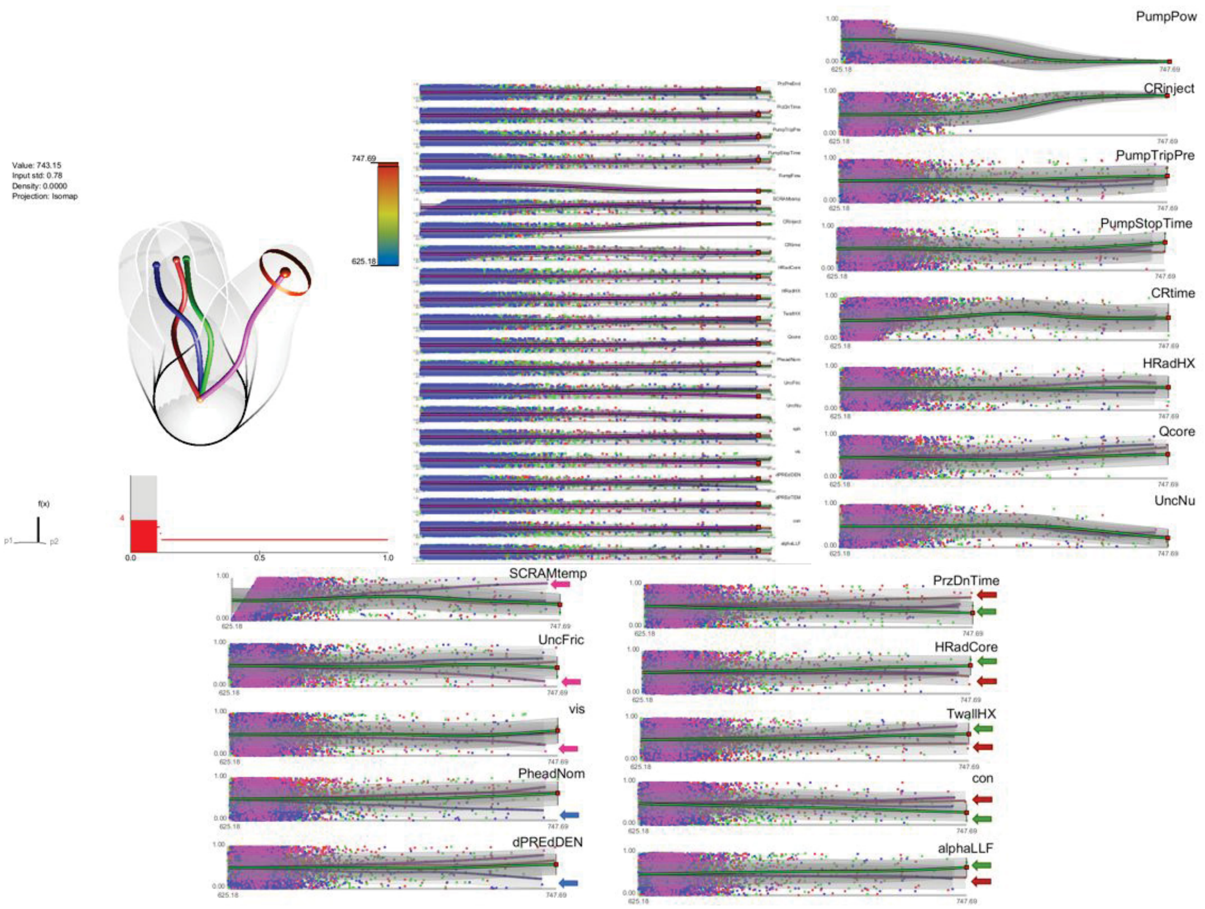


Figure 19: **21D-PCT-All-4C**: Top left: Topological skeleton shown in a 3D view. Top center: inverse coordinate plots for all 21 dimensions showing simulations colored according to cluster id and regression curves of each cluster superimposed per dimension. Top right: inverse coordinate plots for all non-distinguishing parameters (all crystals show similar behavior). Bottom left: the dimensions that distinguish the blue and magenta crystals. Bottom right: The dimensions that differentiate the red and green crystals from one another.

3.7 Conclusions

In this section, we have presented a software tool suitable to analyze and visualize datasets generated by safety analysis codes. In particular, our software could be interfaced with Dynamic PRA algorithms. In such a configuration, a large number of runs of the safety analysis code are performed, where during each run, initial conditions and timing/sequencing of events are changed according to their statistical distribution. For large complex systems, this type of code generates a large amount of data, modeled as high-dimensional scalar functions. Our software is designed to assist the user in the process of extracting useful information from such functions.

This extraction is performed by exploring the correlations between uncertain input parameters and simulation outcome. These correlations are modeled as a Morse-Smale complex and then summarized in various visual display modules, where the reconstructed simulation outcome is directly linked to the input parameters. From an uncertainty quantification point of view, such an analysis tool allows the user to identify and consequently rank variables based on their correlations with simulation outcome (e.g., maximum fuel temperature). Such software becomes even more effective for the analysis of complex systems such as nuclear power plants where the number of variables is very large.

We have described each analysis and visualization module in the system, by explaining the theories and techniques based on a specific application of the software in a 6D demo example involving nuclear safety simulation. Our example is based on code performed on 10000 simulations of a simplified pressurized water reactor (PWR) system for a SCRAM scenario. Our software allows us to identify and visualize the correlations among six parameters and the maximum coolant temperature. We first perform topological analysis to identify the underlying topological structure of the dataset. Statistical information is then summarized and linked to the topological structures extracted from the data, allowing the users to identify correlations between timing/sequencing of events and simulation outcome. We have obtained some intuitive understanding of the structures of such a high-dimensional function, although a more comprehensive interpretation of our results depends on extensive testing and explorations from our collaborators. We further demonstrate the usability of our tool on a 21D case study with more input parameters compared to the 6D case, and obtain similar visualization results. We expect to work closely with the domain scientists and obtain feedback from the end users regarding (a) the interpretation of the testing datasets using our software, (b) the limitations and potential improvements of current analysis techniques, and (c) the potential improvements of the visualization interfaces in terms of usability and interactivity.

4 Adaptive Sampling

4.1 Overview

Nuclear simulations are often computationally expensive, time-consuming, and high-dimensional with respect to the number of input parameters. Thus exploring the space of all possible simulation outcomes, the response surface, is infeasible using finite computing resources. During simulation-based probabilistic risk analysis, it is important to discover the relationship between a potentially large number of input parameters and the output of a simulation using as few simulation trials as possible. This is a typical context for performing adaptive sampling where a few observations are obtained from the simulation, a surrogate model is built to represent the simulation space, and new samples are selected based on the model constructed. The surrogate model is then updated based on the simulation results of the sampled points. In this way, we attempt to gain the most information possible with a small number of carefully selected sampled points, limiting the number of expensive trials needed to understand features of the simulation space.

In this section, we develop a range of topologically-driven adaptive sampling techniques, aimed at understanding the response surface, both locally and globally.

First, we focus on developing topology-driven scoring functions. One critical question that arises in our attempt to build an informative surrogate model is how to sample, both in terms of number and distribution, the input parameter space to yield the most cogent output representation. We explore sampling strategies from the context of what is necessary for reconstructing the topological features of the response surface. We use topological information to guide the selection of sampled points from the candidate set. Topology-driven scoring functions for adaptive sampling selectively choose *topologically informative* points that would be useful in improving the underlying surrogate model, in an iterative fashion.

Second, we analyze the specific use case of identifying the limit surface, i.e., the boundaries in the simulation space between system failure and system success. We explore several techniques for adaptively sampling the parameter space in order to reconstruct the limit surface. First, we seek to learn a global model of the entire simulation space using prediction models or neighborhood graphs and extract the limit surface as an iso-surface of the global model. Second, we estimate the limit surface by sampling in the neighborhood of the current estimate based on topological segmentations obtained locally. Our techniques draw inspirations from topological structure known as the Morse-Smale complex. We highlight the advantages and disadvantages of using a global prediction model versus local topological view of the simulation space, comparing several different strategies for adaptive sampling in both contexts. One of the most interesting models we propose attempt to marry the two by obtaining a coarse global representation using prediction models, and a detailed local representation based on topology. Our methods are validated on several analytical test functions as well as a small nuclear simulation dataset modeled after a simplified Pressurized Water Reactor (PWR).

4.2 Topology-Driven Scoring Functions

Nuclear simulation is typically expensive to evaluate and thus the sampling of the parameter space is necessarily small. As a result, choosing a *good* set of samples at which to evaluate is crucial to glean as much information as possible from the fewest samples. While space-filling sampling designs such as Latin Hypercubes provide a good initial cover of the entire domain, more detailed studies typically rely on adaptive sampling: Given an initial set of samples, these techniques construct a surrogate model and use it to evaluate a *scoring function* which aims to predict the expected gain from evaluating a potential new sample. There exist a large number of different surrogate models as well as different scoring functions each with their own advantages and disadvantages. We have performed an extensive comparative study of adaptive sampling using four popular regression models combined with six traditional scoring functions compared against a space filling design [49]. Furthermore, for a single high-dimensional output function, we introduce a new class of scoring functions based on global topological rather than local geometric information. The new scoring functions are competitive in terms of the root mean square prediction error but are expected to better recover the global topological structure. Our experiments suggest that the most common point of failure of adaptive sampling schemes are ill-suited regression models. Nevertheless, even given well-fitted surrogate models many scoring functions fail to outperform a space-filling design.

4.2.1 Adaptive Sampling Overview

Two main families of prediction models (PMs) are used in the uncertainty quantification (UQ): regression models such as MARS and stochastic models such as Gaussian processes. Our general pipeline as illustrated in Fig. 20 is applicable to both families.

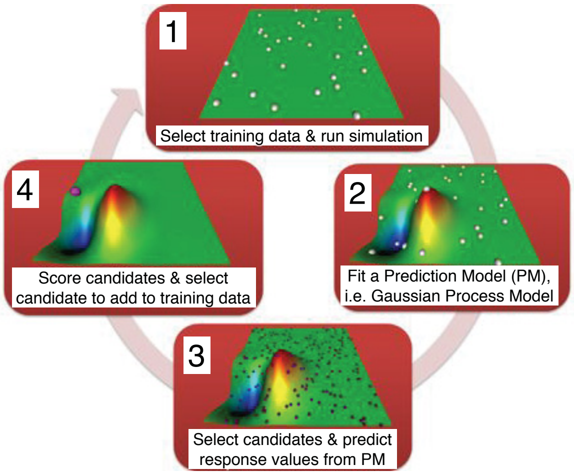

1
Select training data & run simulation
2
Fit a Prediction Model (PM), i.e. Gaussian Process Model
3
Select candidates & predict response values from PM
4
Score candidates & select candidate to add to training data

Figure 20: The iterative pipeline for adaptive sampling. Starting with a set of training points (top) a prediction model is create (right). The model is evaluated at a large number of candidate points (bottom) typically created through a space-filling design. Each candidate point is assigned a score indicating the expected information gain were this point being evaluated. Finally, the point with the highest score is selected and evaluated using the simulation (left). Finally, the new sample is added to the training data and the process is repeated.

We begin by selecting some initial training data, running the simulation and obtaining a collection of true responses at these data points. Second, we fit a prediction model (PM), i.e. a Gaussian Process Model, from the initial set of training data. Third, a large set of candidate points is chosen in the parameter space using Latin Hypercube Sampling (LHS), and the PM is evaluated at these points. It is important to note that we use PM to approximate values at these candidate points, which is highly efficient. Fourth, each candidate point is assigned a score based on some adaptive sampling score functions. Finally, the candidates with the highest scores are selected and added to the set of training data to begin a new cycle.

Traditional score functions are either based on point density, where candidate points which are further away from the existing training data get higher scores; or are based on prediction accuracy, where candidates with higher prediction uncertainty get higher scores. We propose a third class of score functions based on topological information, where candidate points in the area of larger topological changes are assigned higher scores.

4.2.2 Traditional Scoring Functions

We first review several traditional scoring functions. We compare our topological scoring functions to these metrics. There are three main scoring functions, namely, Delta, ALM and EI, as detailed below. Each of them is further augmented with a distance penalization factor, creating three additional scoring functions.

Let $\mathscr{T} = \{\mathbf{z}\}_{i=1}^{n}$ be the set of n training points in dimension d. The true response at a point $\mathbf{z} \in \mathscr{T}$ is denoted as $y(\mathbf{z})$. Let $\mathscr{S} = \{\mathbf{x}\}_{i=1}^{m}$ be the set of m candidate points in dimension d. The predicted response at a point $\mathbf{x} \in \mathscr{S}$ is denoted as $\hat{y}(\mathbf{x})$.

Delta. This criterion can be seen as a way to evenly sample the range space of the function. It is defined as the absolute value of the difference between the predicted response at a candidate point and the response at the nearest training point. Points are chosen wherever the response model predicts a large gap in function value. Note that while the delta criterion is very intuitive it does not consider either gradient magnitudes nor "predictability". For example, a point in the middle of a steep but linear ramp is easily predicted even though it may have a large difference in function value. Similarly, a point with a large difference in function value far away from the nearest sample may not be as interesting as a slightly smaller difference in a highly sampled region. Formally, for a point $\mathbf{x} \in \mathscr{S}$, $Delta(\mathbf{x})$ is the absolute difference, to the q-th power, between the predicted response and the response observed at the closest point in the training sample, as measured by the L_p distance metric, that is, $d(\mathbf{x},\mathbf{x}') = (\sum_{i=1}^{d} |x_i - x_i'|^p)^{1/p}$. That is, for some fixed parameters p and q, let $\mathbf{x}^* = \arg\min_{\mathbf{z}\in\mathscr{T}} d(\mathbf{x},\mathbf{z})$, then $Delta(\mathbf{x}) = |\hat{y}(\mathbf{x}) - y(\mathbf{x}^*)|^q$. In the default setting, $p = 2$ and $q = 2$.

ALM. This is the Active Learning MacKay criterion described in [46] which attempts to optimize the predictive variance. The idea is that the variance represents a notion of uncertainty in the prediction and new samples should be evaluated in the least well understand regions of the parameter space. For the GPM the variance can be computed directly from the model which appears to be a significant advantage. For the other prediction models we use bootstrapping to estimate the variance.

EI. This is the expected improvement criterion. This can be seen as a combination of the expected prediction error used in the ALM method and the Delta criterion. Points are chosen that either show a large uncertainty in their current prediction or have a large discrepancy with the closest existing sample. Our predication model uses $EI(\mathbf{x}) = (|\hat{y}(\mathbf{x}) - y(\mathbf{x}^*)|^2 + ALM(\mathbf{x}))^{1/2}$.

Distance Penalization. Each of the above three scoring functions can be augmented with a distance penalization factor, therefore creating three additional scoring functions, namely, *DeltaDP*, *ALMDP* and *EIDP*. For DeltaDP, the Delta criterion with an additional penalty term, we can prevent samples from lying too close to the training set. The scaling attempts to balance the goal of sampling in areas of large function variance with the ability to detect yet unknown features by preferring under-sampled areas. For a point $\mathbf{x} \in \mathscr{S}$, $DeltaDP(\mathbf{x}) = Delta(\mathbf{x}) * \rho_{\mathbf{x}}$, where $\rho_{\mathbf{x}}$ is the distance scaling factor. Recall $d_{\mathbf{x}} = d(\mathbf{x}, \mathbf{x}^*)$ is the distance from $\mathbf{x}$ to the closest point in the training data, and D is a distance vector of $d_{\mathbf{x}}$ for all $\mathbf{x} \in \mathscr{S}$. $\rho_{\mathbf{x}} = \rho_{\mathbf{x}}(d_{\mathbf{x}}, d_0, p_0)$, where d_0 is the range and q_0 is the quantile (by default, d_0 is the q_0 quantile of D). If $d_{\mathbf{x}} > d_0$, set $\rho_{\mathbf{x}} = 1$, otherwise $\rho_{\mathbf{x}} = 1.5 d_{\mathbf{x}} - 0.5 d_{\mathbf{x}}^3$, where the coefficients are taken from spherical semivariogram. Similarly we define $ALMDP(\mathbf{x}) = ALM(\mathbf{x}) * \rho_{\mathbf{x}}$, where we approach a more space-filling point selection; and $EIDP(\mathbf{x}) = EI(\mathbf{x}) * \rho_{\mathbf{x}}$. By default we use $q_0 = 0.5$.

4.2.3 Topological Scoring Functions

All of the scoring functions discussed above pick sample points more or less directly based on the idea of globally improving the prediction accuracy. However, these points are not necessarily the optimal candidates. Imagine, for example, a steep mountain that (by random chance) has already been sampled both close to its peak as well as somewhere near the base. For points on the slope of the mountain, the prediction will show a large difference in function value thus making it attractive for most standard techniques. However, evaluating the prediction in more detail would also show that even taking a sizable prediction error into account the global structure of the mountain would not change by adding a point on its slope. More specifically, the single mountain would remain a single mountain for a wide range of potential new values even considering errors and uncertainty. This rationale leads to topology based scoring function aimed at discovering the global structure – the topology – of a function rather than its detailed geometry. In particular, we propose three different topology based scoring functions, named, *TopoHP*, *TopoP* and *TopoB* as detailed below, illustrated in Fig. 21.

TopoHP. The first strategy is aimed at sampling at or near predicted critical points with significant influence on the topology. It is defined as the persistence of a candidate point within an (approximated) Morse-Smale complex constructed from oversampling the current response model. Given the current response model, we evaluate its prediction at all candidate points and compute the Morse-Smale complex of the resulting point set by combining both training points and candidate points. We then assign all critical points of the complex that are part of the candidate sets their persistence as score and assign a zero score to all regular points within the candidate sets. Referring to

Fig. 21(a), the silver points illustrate the training points $\mathscr{T}$ and the purple points correspond to the candidate points $\mathscr{S}$. We construct a Morse-Smale complex over $\mathscr{T} \cup \mathscr{S}$, and return the persistence of the critical points within the candidates. Here, point $\mathbf{x}$ is selected with the highest persistence, therefore, the highest $TopoHP(\mathbf{x})$.

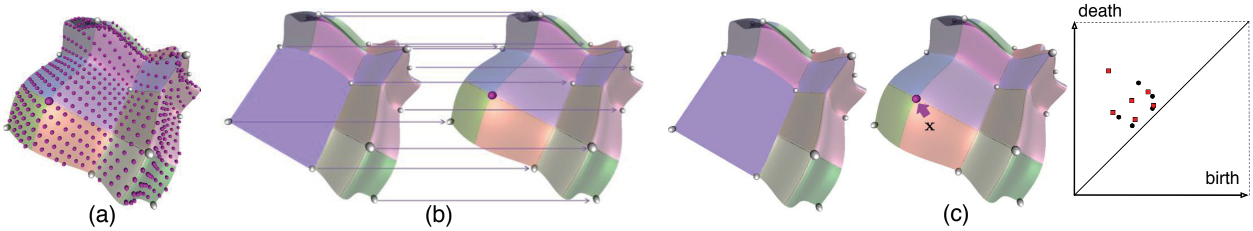


Figure 21: (a) TopoHP: Construct a Morse-Smale complex from training data (silver) as well as all candidates with predicted responses (purple), return the persistence of the critical points within the candidates. Point $\mathbf{x}$ is selected with the highest $TopoHP(\mathbf{x})$. (b) TopoP: average change in persistence for all extrema before (left) and after (right) inserting a candidate $\mathbf{x}$ into the Morse-Smale complex. (c) Morse-Smale complexes before (left) and after (middle) inserting a candidate point $\mathbf{x}$; TopoB (right): bottleneck distance between the corresponding persistence diagrams before (circles) and after (squares) insertion.

TopoP. Similar to a bootstrapping approach, this strategy aims to evaluate how much the topology (as represented by the persistences) would change if a new candidate point is added. It is defined as the average change of persistence for all current extrema when a given candidate point with its predicted response is inserted into the Morse-Smale complex. As shown in Fig. 21(b), we first construct the Morse-Smale complex of all training data $\mathscr{T}$ (silver points). Then for each candidate point $\mathbf{x} \in \mathscr{S}$, we construct a new Morse-Smale complex consisting of $\mathscr{T} \cup \mathbf{x}$ (Fig. 21(b) right). To score the candidate $\mathbf{x}$, we compute the change in persistence for each training point $\mathbf{x} \in \mathscr{T}$ that remains as an extrema point between the original and enhanced Morse-Smale complex, and average these changes to obtain a single, nonnegative value.

TopoB. For each point $\mathbf{x} \in \mathscr{S}$, $TopoB(\mathbf{x})$ is defined as the bottleneck distance between the persistence diagram of the Morse-Smale complex over $\mathscr{T}$ versus the Morse-Smale complex consisting of $\mathscr{T} \cup \mathbf{x}$. This strategy is similar to the TopoP scoring except that the bottleneck distance not only takes the persistence values into account but also the order and nesting of the corresponding simplification. This is shown in Fig. 21(c), where left and middle illustrate the (approximated) Morse-Smale complexes before and after inserting a candidate point $\mathbf{x}$, and right displays the corresponding persistence diagrams of these complexes. $TopoB(\mathbf{x})$ is defined as the bottleneck distance between them.

4.2.4 Discussions

After running an extensive set of experiments in two to five dimensions with nine different scoring functions (where the results are detailed in [49]), some general trends appear even though the fundamental question of which particular scoring function to use remains largely inconclusive. The results given in the section is only a small first step in the this direction. The first, not necessarily surprisingly result is that the quality of the underlying regression model plays a key role in the performance of any adaptive sampling technique. In this study the GPM model performs the best and is the only one showing significant differences between scoring functions. Overall, it seems that the combination of ALM and GPM is the preferred choice even though some functions perform well with the Delta criteria.

The remaining regression models all show problems fitting many of the test functions and the large variability suggest that they are sensitive to specific sample locations. In the cases where reasonable fits have been achieved all scoring functions perform equally well (or poorly).

The new topological scoring functions are largely competitive in terms of RMSE and often perform among the top scoring functions. Some results suggest that detailed fits with a high number of samples are less well suited for topological scoring functions as they are designed to recover larger scale features. Nevertheless, topological scoring functions are expected to better recover the global structure of a function, and finding quantitative metrics to test this hypothesis as well as expanding the class of such functions will be the focus of future research.

4.3 Adaptive Sampling Models to Recover the Limit Surface

To study the reliability and safety of complex systems (e.g. nuclear power plants, airplanes, chemical plants), one typically performs a combination of event tree and fault tree analysis [67]. However, these methods are characterized by the following disadvantages: (a) The timing of events is not explicitly modeled; (b) The ordering of events is preset by the analysts; (c) The modeling of complex accident scenarios is driven by expert judgment. For these reasons, there is currently an increasing interest in the development of dynamic Probabilistic Risk Assessment methodologies (DPRA) [63] since they can be used to address the deficiencies of the conventional methods listed above.

For DPRA, propagation of uncertainties in the analysis of complex systems employs sampling algorithms which perform a series of simulation runs given a set of uncertainty parameters. However, typically the set of uncertain parameters is very large and the computational cost of each run is very high. Consequently, the space of the possible solutions, the response surface, can be sampled only very sparsely and this precludes the ability to fully analyze the impact of uncertainties on the system. For safety analysis applications, the following points often emerge: (a) Many regions of the response surface are not of interest; (b) The set of parameters that are of safety concern is a small subset of the original set of uncertainty parameters.

Currently, the class of most used sampling algorithms include: classic stochastic sampling (e.g., Monte-Carlo [35,44], stratified sampling such as Latin hypercube sampling [36], importance sampling [30] and orthogonal arrays based [59] algorithms) and deterministic algorithms (e.g., Polynomial Chaos Expansions [58] and Quasi Monte-Carlo [7]). However, all of these sampling algorithms do not possess a sampling strategy that explicitly takes into account the results of previous simulations. Adaptive sampling algorithms, on the other hand, adapts a sampling strategy that chooses the next sample based on the results obtained by previous samples through a statistical learning models and, thus, focus sampling in risk sensitive regions such as boundaries between system safe and system failure: the limit surface.

We present here three adaptive sampling models that could be used to infer the behavior of system by estimating the limit surface. All three models share a general adaptive sampling pipeline but differ from one another in terms of the candidate selection process. The first model relies on globally probing a prediction model to estimate the limit surface as an iso-surface of the global response surface. The second model is topology-based. It improves upon the first model by taking into account the topology of the prediction model based on its Morse-Smale complex structure, and samples points in local areas that are considered topologically relevant. The final model is designed to be completely data-driven. It computes a candidate set directly from the neighborhood graph constructed from the training data, without any dependencies on a particular prediction model. We discuss the advantages and limitations of our proposed models and demonstrate their fidelity on several synthetic datasets as well as a small nuclear simulation dataset modeled after a simplified PWR.

4.3.1 Adaptive Sampling Models

We introduce three different adaptive sampling models. They share a general pipeline but differ from one another in terms of the candidate selection process. In the first model $\mathbf{M_1}$, we rely on globally probing a prediction model (PM), e.g. a Gaussian process model (GPM), to estimate a candidate set that lies on the limit surface. The second model $\mathbf{M_2}$ improves upon this technique by computing the topology of the PM and sampling in areas deemed interesting by the topology rather than querying the entire domain. The final model $\mathbf{M_3}$ computes a candidate set directly from the training data without the use of a PM.

The general adaptive sampling pipeline (Figure 22 top left) begins by selecting some initial training data, running the simulation and obtaining a collection of true responses at these data points. Second, it fits a response surface surrogate model from the initial set of training data. Third, a set of candidate points is chosen in the parameter space based on certain sampling technique, and the surrogate model is evaluated at these points, obtaining a set of approximated values. Fourth, each candidate point is assigned a score based on some adaptive sampling scoring function (usually derived from qualitative or quantitative relations between the training points, their true and estimated response values). Finally, the candidate(s) with the highest score(s) are selected and added to the set of training data to begin a new cycle.

Before we describe each of the three models in detail, we describe their common ingredients. Two main families of PMs are used in the uncertainty quantification: regression models such as Multivariate Adaptive Regression Splines (MARS) and stochastic models such as Gaussian processes. Both $\mathbf{M_1}$ and $\mathbf{M_2}$ uses an (easily replaceable) PM, i.e. a Gaussian process model (GPM), as the surrogate model. We use a specific implementation known as the Sparse Online Gaussian Process [17]. To sample from a PM, we employ a central Voronoi tessellation (CVT) scheme [20]. CVT samples cover the domain relatively densely but are sparser than points obtained by Latin Hypercube Sampling (LHS). All three models select a set of candidate points by locating the current

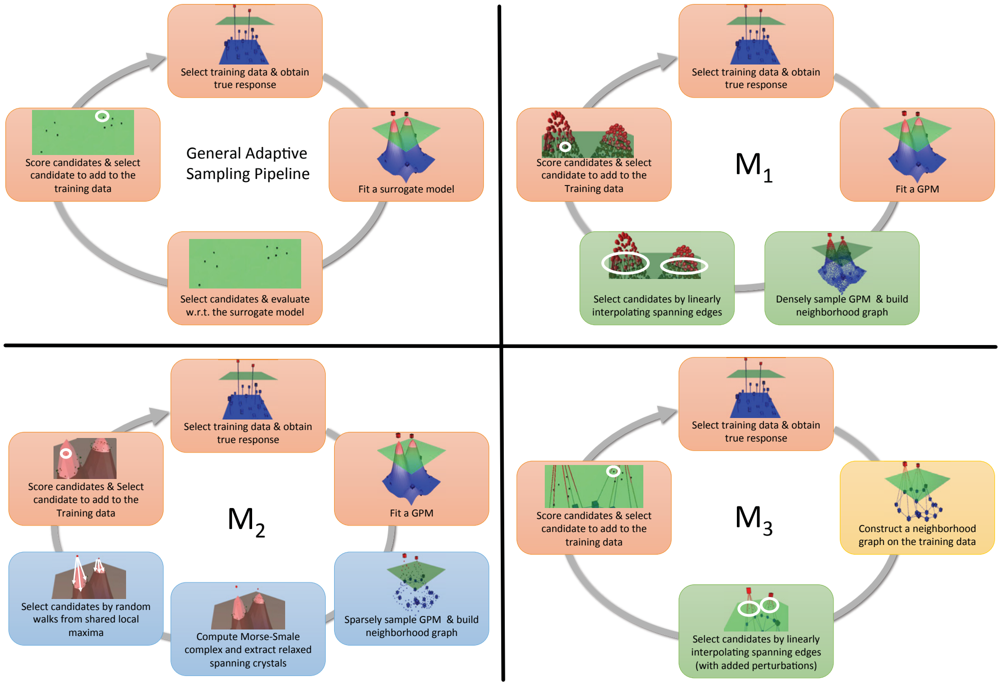

Select training data & obtain true response
Score candidates & select candidate to add to the training data
General Adaptive Sampling Pipeline
Fit a surrogate model
Select candidates & evaluate w.r.t. the surrogate model
Select training data & obtain true response
Score candidates & select candidate to add to the Training data
M1
Fit a GPM
Select candidates by linearly interpolating spanning edges
Densely sample GPM & build neighborhood graph
Select training data & obtain true response
Score candidates & Select candidate to add to the Training data
M2
Fit a GPM
Select candidates by random walks from shared local maxima
Compute Morse-Smale complex and extract relaxed spanning crystals
Sparsely sample GPM & build neighborhood graph
Select training data & obtain true response
Score candidates & select candidate to add to the training data
M3
Construct a neighborhood graph on the training data
Select candidates by linearly interpolating spanning edges (with added perturbations)

Figure 22: Pipelines for the general adaptive sampling model and the new models: $\mathbf{M_1}$, $\mathbf{M_2}$ and $\mathbf{M_3}$.

estimate of the limit surface. The limit surface is estimated by imposing a neighborhood graph on all points with known or estimated observations and linearly interpolating along edges that span the threshold value. In the case of points estimated with a GPM, their mean value predictions are used. While many neighborhood graphs are possible [16,48], we use the *relaxed Gabriel graph* [5] which has been shown to have relatively dense connectivities. In terms of the chosen adaptive sampling scoring function, we employ one based on point density, referred to as the *point density criterion*, where candidate points which are further away from the existing training data get higher scores. Other scoring criteria could also being employed (e.g. Active Learning MacKay criterion [46] or expected improvement criterion [40, 43, 62]), arguably leading to different convergence behaviors. However, since our focus in this section is not on scoring function design but model selection, we employ one of the simplest scoring functions.

Finally, we need a few concepts and notations. Suppose t is the function value threshold that corresponds to the limit surface and without loss of generality, suppose the failure region is considered part of the response surface whose function value is above t. An edge in a relaxed Gabriel graph is a *spanning edge* if it spans the threshold value t in the range space, in other words, if its endpoints have function values on the opposite sides of t. A Morse-Smale crystal is a *spanning crystal* (i.e. spanning the limit surface) if its local maximum has a function value above t and its local minimum has a function value below t. A Morse-Smale crystal is a *relaxed spanning crystal* if its local maxima has a function value above $t - \delta$ and its local minimum has a function value below $t + \delta$, for some limit surface tolerance parameter $\delta \geq 0$. In a nutshell, relaxed spanning crystals allow us to investigate regions from the response surface of a PM that are on or near the current estimate of the limit surface, thus are important for exploratory analysis. This would become clearer subsequently when $\mathbf{M_2}$ is described in detail.

$\mathbf{M_1}$: Global PM Limit Surface Recovery. As illustrated in Figure 22 top right, we begin with a set of training points and build a PM model using the GPM. Second, we *densely* sample the GPM over the entire domain using a CVT scheme, and construct a relaxed Gabriel graph on the union of these CVT samples with the training points. Third, candidate points are generated by linearly interpolating at the threshold value t along each spanning edge (i.e. edge in the graph that spans the threshold value t in the range space). Fourth, the set of candidate points are ranked based on the point density criterion, where a candidate with the highest score is selected and added to the set of training points to begin a new cycle.

$\mathbf{M_2}$: Topology-Based PM Limit Surface Recovery. As illustrated in Figure 22 bottom left, $\mathbf{M_2}$ improves the global PM query by extracting the topology of the PM and focusing samples in specific regions. First, we fit a GPM on the initial set of training points. Second, we *sparsely* sample the GPM over the entire domain using a CVT scheme. Sparsity means that we query just enough points to obtain a reasonable estimation of the topology of the GPM. Then we impose a relaxed Gabriel graph on the union of these CVT samples with the training points, and approximate a Morse-Smale complex based on the graph structure. Third, by examining the local extrema of each crystal in the Morse-Smale complex, we can quickly discard those crystals that are far away from the limit surface, and keep only the crystals that are located in regions of interests. The remaining crystals are precisely the relaxed spanning crystals defined previously, which are on or near the limit surface of the current surrogate model. Fourth, we store the local maxima of all relaxed spanning crystals. From each such maxima (which are considered in the failure region with high probability), we employ some sampling techniques (i.e. random walks) in its local neighborhood to obtain a set of candidate points that are near the boundary of the failure region. Those points are determined based on the GPM such that their predicted function values are in the range $[t-\delta, t+\delta]$ for some limit surface tolerance parameter $\delta \geq 0$. Finally, the set of candidate points collected across all such local maxima are ranked and selected the same way as in $\mathbf{M_1}$ to begin a new cycle.

To sample the neighborhood of a local maxima shared among some relaxed spanning crystals, we perform random walks described below. Starting from the maxima, we perform walks in enough random directions (i.e. scale with dimension) to characterize the domain space. The path of a random walk terminates when it hits the domain boundary or it reaches a point whose predicted function value from the GPM lies within $[t-\delta, t+\delta]$. As the distance between the random walks increases as we move further away from the seeding maxima, we add new seed points and increase the number of paths at fixed intervals along each path to better cover the domain.

In comparison with $\mathbf{M_1}$, we expect several advantages of $\mathbf{M_2}$ with added benefits coming from the topological knowledge of the surrogate model. First, $\mathbf{M_2}$ allows us to more densely sample the interesting regions and quickly reject uninteresting regions of the domain space. One could achieve a similar result by refining the sampling of $\mathbf{M_1}$ near the estimated limit surface, but this would require more samples to be generated from an already densely sampled PM. Second, $\mathbf{M_2}$ introduces a tolerance parameter δ in defining the relaxed spanning crystals, therefore allowing exploratory analysis of potential failure regions within a controlled level of uncertainty. Overall, as we demonstrate in Section 4.3.2, $\mathbf{M_2}$ behaves comparably with $\mathbf{M_1}$ under general settings, and for certain special scenarios it performs drastically better.

$\mathbf{M_3}$: Data-Driven Limit Surface Recovery. As opposed to building a PM, $\mathbf{M_3}$ (Figure 22 bottom right) begins by directly building a neighborhood structure as the surrogate model (e.g. a relaxed Gabriel graph) on the initial training data. It then creates a candidate set by first obtaining linearly interpolated points at t along spanning edges of the graph, and introducing a random perturbation ε along all dimensions to these points. Finally, these candidates are ranked and selected in a similar fashion as in $\mathbf{M_1}$ and $\mathbf{M_2}$ to begin a new cycle.

The candidate set we obtained through $\mathbf{M_3}$ is arguably sparser, however introducing a certain amount of perturbation to the linearly interpolated points enables us to explore the region surrounding the limit surface further. Note that it is not necessary to introduce this type of perturbation to candidates obtained from linear interpolation in $\mathbf{M_1}$ since it relies on a PM which always queries a new set of sampled points during each round, resulting in drastically different graph structures. On the other hand, since $\mathbf{M_3}$ does not use a PM, the graph obtained during each round changes only slightly, such that without a random perturbation the candidate points are generally located linearly along the edge of the graph, which is less desirable.

4.3.2 Experiments

We demonstrate our three adaptive sampling models on six two-dimensional datasets with specially prescribed threshold values to exploit their limit surfaces, which are the boundaries of failure regions of varying sizes and shapes. We start with several synthetic datasets (with analytic closed forms) of various complexity, and end with a nuclear dataset modeled after a simplified Pressurized Water Reactor (PWR). It is important to note that our proposed models could be extended and applied to high-dimensional datasets, although it becomes much harder to validate the results through visual inspection.

Our first three synthetic datasets are generated as mixtures of Gaussians: **MixtureA**, **MixtureB** and **MixtureC**. The fourth and fifth synthetic datasets are based upon a modified distance field for **DistanceD** and an inverted modified distance field for **DistanceE**. The last nuclear dataset **NuclearF** comes from a simplified PWR model that has been used for station blackout analysis. We now describe the parameter settings for the experiments performed. The domain of each dataset has been normalized to be $[0,1] \times [0,1]$. For the initial training points, we use the same 10 points for **MixtureA** and **MixtureB**, and the same 20 points for the remaining datasets. We add additional 100 adaptively sampled points (one at a time) to the training points through our experiments. To recover points that lie on the true limit surface, we sample 10K CVT samples from the ground truth model (either in closed form for the synthetic datasets or in terms of black-box simulation for the nuclear dataset), impose a relaxed Gabriel graph on these samples and linearly interpolate the spanning edges at the given threshold value. Similar operations are performed to recover points that lie on the estimated limit surface for models $\mathbf{M_1}$ and $\mathbf{M_2}$. Since $\mathbf{M_3}$ does not rely on a PM, thus not allowing CVT sampling of the domain, the set of points that lies on the estimated limit surface is obtained by linearly interpolating the spanning edges of the relaxed Gabriel graph constructed on the current training points. $\mathbf{M_2}$ introduces a limit surface tolerance parameter δ, which is typically set to 5% of the range of the function (except for $\delta = 10\%$ for **MixtureC** and $\delta = 20\%$ for additional tests run on **DistanceD**). $\mathbf{M_3}$ employs a perturbation parameter ε, which is set to be 5% of the length of each dimension in the domain space (in the case where each dimension is normalized to be $[0,1]$, $\varepsilon = 0.05$). The limit surface threshold values are 0.6 for **MixtureA**, **MixtureB**, **MixtureC** and **DistanceD**; 2.58 for **DistanceE**; 800 for **NuclearF**. For $\mathbf{M_1}$, we need to sample the GPM densely to recover the points on or near the estimated limit surface, therefore we use 2000 CVT samples. On the other hand, for $\mathbf{M_2}$, we only need to sample the GPM sparsely, using just enough samples to recover its topology, thus we query 200 CVT samples.

4.3.2.1 Mixture of Gaussians

We start with three functions generated as mixtures of Gaussians. As shown in Figure 23, in all three cases, each model is able to extract each component of the limit surface to varying degrees of fidelity.

The first dataset **MixtureA** is sampled from a single Gaussian function centered in the domain, given by the function, $z = e^{-\frac{(x-0.5)^2+(y-0.5)^2}{0.2}}$. Its limit surface is the boundary of a failure region that covers a large portion of the interior of the domain and contains a single connected component. The limit surface is smooth and circular with uniform curvature. Therefore it serves as a trivial validation dataset as most models should recover it with good accuracy and with few adaptively sampled points. As shown in Figure 23 **MixtureA**, all three models succeed in this regard.

The second dataset **MixtureB** adds complexity by adding a second component and moving the failure regions such that they intersect with the domain boundary. The function used for this is just the direct combination of two uniform Gaussian kernels: $z = e^{-\frac{x^2+y^2}{0.3}} + e^{-\frac{(x-1)^2+(y-1)^2}{0.3}}$. It is interesting to see in Figure 23 **MixtureB** that both $\mathbf{M_1}$ and $\mathbf{M_2}$ estimate similar limit surfaces with inaccuracies near the domain boundary, while $\mathbf{M_3}$ recovers the limit surface with a more accurate geometry.

The third dataset **MixtureC** distorts the shape of the Gaussian peaks to be ellipsoidal and places two of the Gaussian centers close together to form a more complicated shape for one component of the failure region. In addition, another Gaussian function is centered nearby and has a saddle point near the limit surface. A fourth Gaussian function is created whose local maxima lies just below the limit surface. The closed form equation is given by: $z = 0.5e^{-\frac{(x-0.2)^2}{0.02}+\frac{(y-0.2)^2}{0.02}} + e^{-\frac{(x-0.3)^2}{0.02}+\frac{(y-0.7)^2}{0.03125}} + 0.8e^{-\frac{(x-0.7)^2}{0.03125}+\frac{(y-0.7)^2}{0.03125}} + 0.8e^{-\frac{(x-0.8)^2}{0.005}+\frac{(y-0.45)^2}{0.02}}$. In Figure 23 **MixtureC**, both $\mathbf{M_1}$ and $\mathbf{M_3}$ seem to capture the geometry of the right component of the limit surface better than $\mathbf{M_2}$.

For **MixtureC**, since the current limit surface tolerance parameter δ is set to be 10% of the range space, neither the saddle or the lowest peak (see Figure 23 Mixture C (a)) are deemed interesting (that is, considered as part of the relaxed spanning crystals), therefore no candidate points are placed in their local neighborhood. As we will see in Section 4.3.2.2, in certain situations, varying δ allows exploration of the domain space that may lead to identifications of new failure regions that are otherwise undetected by the global PM.

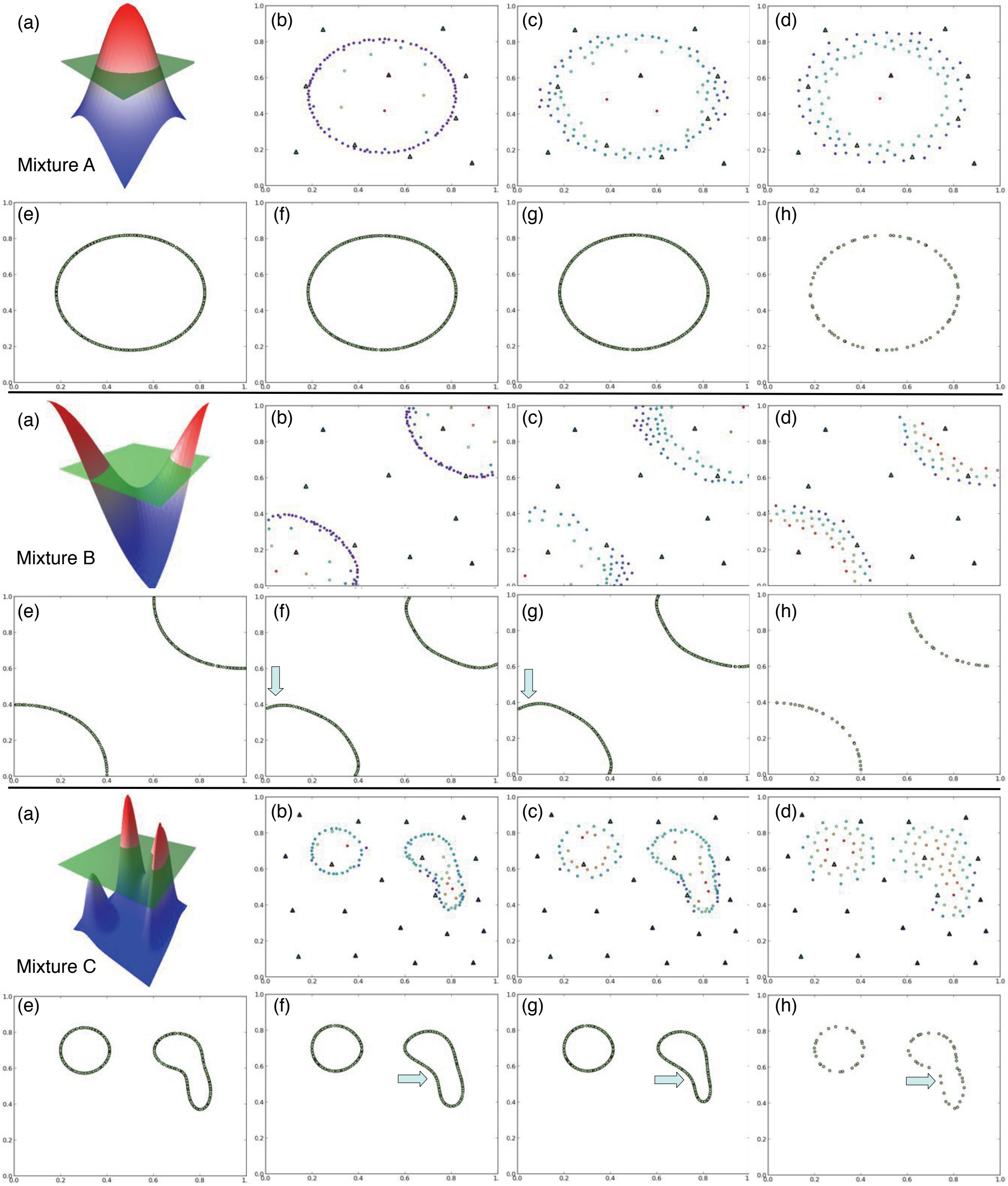


Figure 23: From top to bottom: Synthetic datasets **MixtureA**, **MixtureB** and **MixtureC**. For each dataset: (a) Surface rendering of the true response model and the true limit surface; Adaptively sampled points and training points at the end of the process under (b) $\mathbf{M_1}$, (c) $\mathbf{M_2}$ and (d) $\mathbf{M_3}$; Limit surfaces extracted from (e) the true response model, (f) $\mathbf{M_1}$, (g) $\mathbf{M_2}$ and (h) $\mathbf{M_3}$. Initial training (adaptive sampled) points are solid triangles (circles), colored by true (estimated) responses.

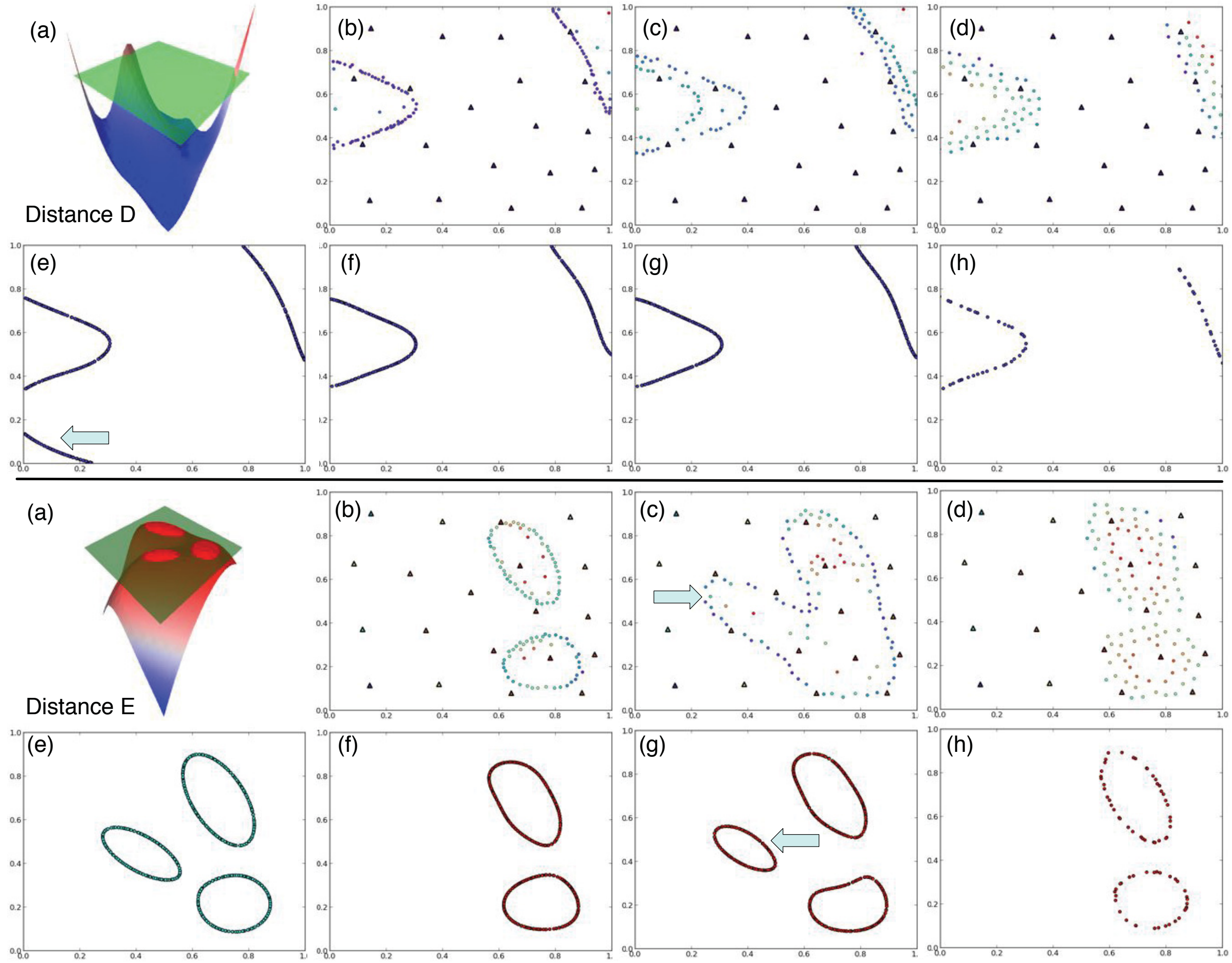


Figure 24: Synthetic datasets **DistanceD** (top) and **DistanceE** (bottom). For each dataset: (a) Surface rendering of the true response model and the true limit surface; Adaptively sampled points and training points at the end of the process under (b) $\mathbf{M_1}$, (c) $\mathbf{M_2}$ and (d) $\mathbf{M_3}$; Limit surfaces extracted from (e) the true response model, (f) $\mathbf{M_1}$, (g) $\mathbf{M_2}$ and (h) $\mathbf{M_3}$. Initial training (adaptive sampled) points are solid triangles (circles), colored by true (estimated) responses.

4.3.2.2 Distance Field and Inverted Distance Field

To avoid performance bias of GPM on certain types of functions (e.g. mixture of Gaussians), it is important to construct test datasets that are geometrically distinct, as shown in Figure 24. We start with a dataset **DistanceD** constructed from a modified distance field. Such a function is composed of five centers and to each center we assign an anisotropic covariance matrix. The function value at any location is given as an L_p-norm over the vector of (anisotropic) distances to each of the centers. Therefore, each of the five center points defines a local minimum. However due to the interactions between the different centers, it produces geometrically different shapes than the ellipses, or combinations of ellipses, given by the Gaussian family of functions. The function in closed form is expressed as $f(\vec{x}) = \left(\sum_i \langle \vec{x} - \vec{c}_i, A_i(\vec{x} - \vec{c}_i) \rangle^p\right)^{1/p}$, where $\vec{c}_i$ is the center point, A_i is the covariance matrix and the parameter p allows tuning the smoothness of this approximation to a non-differentiable distance field ($p = \infty$ gives a true distance field). We set $p = 2$ here.

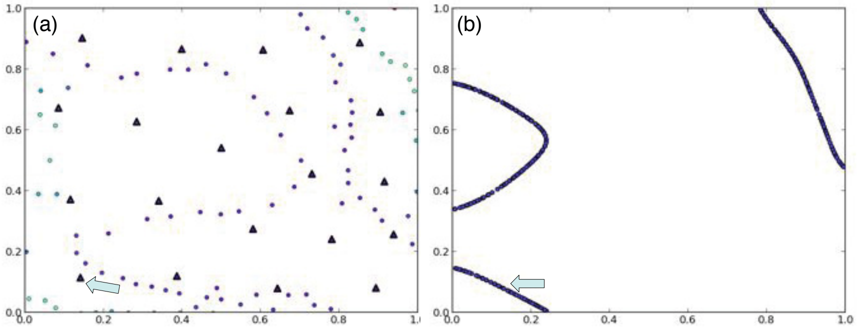


Figure 25: $\mathbf{M_2}$ for **DistanceD**: Increasing limit surface tolerance parameter δ from 5% to 20% of the range space allows us to accurately capture the third component of the limit surface even though we have no initial training data whose response is above the threshold for this region.

As shown in Figure 24 **DistanceD**, all three models fail to recover the third component of the limit surface (pointed by cyan arrow). This is due to the fact that the initial training points in or near that region have function values lower than the threshold, therefore the GPM from $\mathbf{M_1}$/$\mathbf{M_2}$ and the relaxed Gabriel graph from $\mathbf{M_3}$ fail to obtain a fit that predicts a potential interesting region surrounding that location. On the other hand, by increasing the tolerance level from 5% to 20% of the range space, we allow further exploration of the response surface. Subsequently our topology-based $\mathbf{M_2}$ is able to extract the third component successfully, as shown in Figure 25. This is our first example illustrating the potential exploratory power of a topology-based adaptive sampling model, where the parameter δ measures the amount of uncertainty.

Finally, our last synthetic dataset **DistanceE** is constructed by negating (i.e. inverting) the output values of the functional form defined in **DistanceD**, adjusting p to be 3, and deriving new centers with new covariance matrices. A limit surface threshold value that leads to three connected components in the failure region is prescribed. As shown in Figure 24 **DistanceE**, only the topology-based $\mathbf{M_2}$ manages to recover all three components of the limit surface using a tolerance threshold of 5% in the range space. This is our second example showcasing that the tolerance parameter δ allows $\mathbf{M_2}$ to detect areas near the limit surface and place candidate points in that region, potentially enhancing the exploratory power of the model.

4.3.2.3 Nuclear Simulator

To place our techniques in the appropriate setting, we test the validity of our proposed adaptive sampling models for a PRA application of nuclear systems, in which a two-dimensional simplified PWR simulation has been implemented using Simulink [51]. It considers a simplified PWR primary loop including the following systems: automatic depressurization, high pressure injection system (HPIS), low pressure injection systems (LPIS) and diesel generators (DG). The scope of this analysis is to evaluate the impact of Loss Of Offsite Power (LOOP) on the maximum fuel temperature. In particular, domain scientists aim to analyze uncertainties associated with the time at which the power grid is disconnected from the plant and the time associated with the diesel generators starting successfully. There are two input variables of interests in the simulation: T_{PG_SD} and T_{DG}, and the output variable is the core temperature. A typical scenario is as follows (see Figure

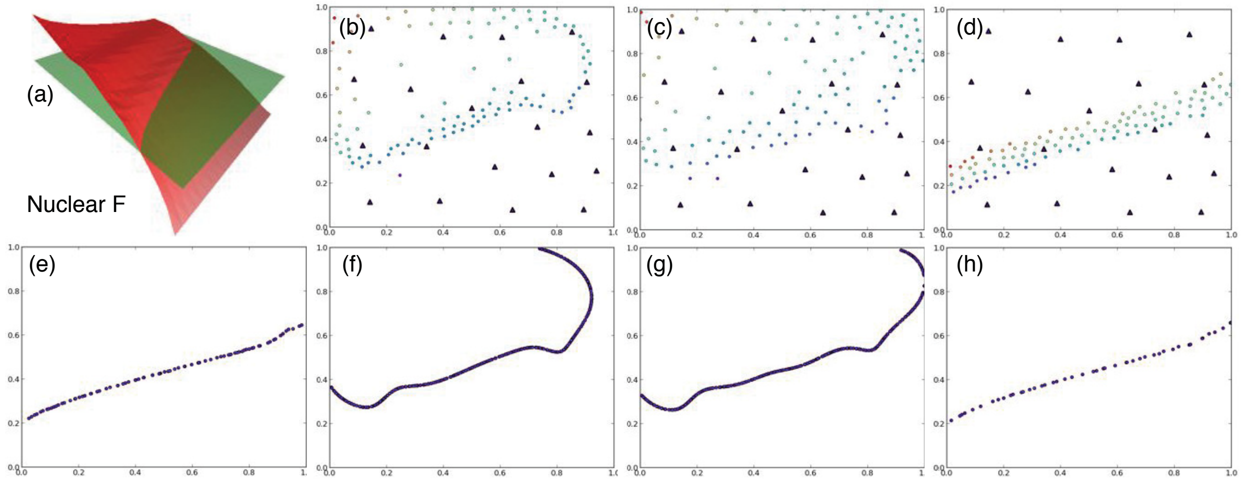


Figure 26: Nuclear dataset **NuclearF**. (a) Surface rendering of the true response model and the true limit surface; Adaptively sampled points and training points at the end of the process under (b) $\mathbf{M_1}$, (c) $\mathbf{M_2}$ and (d) $\mathbf{M_3}$; Limit surfaces extracted from (e) the true response model, (f) $\mathbf{M_1}$, (g) $\mathbf{M_2}$ and (h) $\mathbf{M_3}$. Initial training (adaptive sampled) points are solid triangles (circles), colored by true (estimated) responses.

27):

- External events force the reactor trip at time 0;
- Offsite power is no longer available at T_{PG_SD} (LOOP condition);
- Diesel generators fail to start (station blackout condition) and batteries provide energy only for instrumentation;
- Temperature of the core starts to rise since heat removal is incapacitated. A failure condition is reached when temperature reach 800C;
- Diesel generators become available T_{DG} after LOOP condition and the ECCS is now able to remove decay heat.

For the above dataset named **NuclearF**, the results of the adaptive sampling models are shown in Figure 26. It is interesting to note that $\mathbf{M_3}$ manages to recover the most accurate estimate of the limit surface. In the situation where the end users do not have enough prior information to know the *right* choice of PM to impose on the data, $\mathbf{M_3}$ could become extremely valuable since it does not impose the bias from the PM (as the GPM has done in this case). However, $\mathbf{M_3}$ ignores areas of the domain without any training data, therefore it is not a good exploration but a good exploitation model. Both $\mathbf{M_1}$ and $\mathbf{M_2}$ explore more of the domain space, at the cost of wasting sample points in potentially uninteresting regions. The catch here is that if we do not know what regions are interesting, it might be good to explore the domain for validation and verification purposes through $\mathbf{M_1}$ or $\mathbf{M_2}$.

To further illustrate the bias GPM in $\mathbf{M_1}$ and $\mathbf{M_2}$ impose on the data which lead to less desirable limit surface estimates in Figure 26(f) and (g), we illustrate the evolution of the estimated response surfaces as well as estimated limit surfaces for all three models in Figure 28. These surfaces are rendered after every 10 adaptively sampled points are added to the training data up to 90 points. It

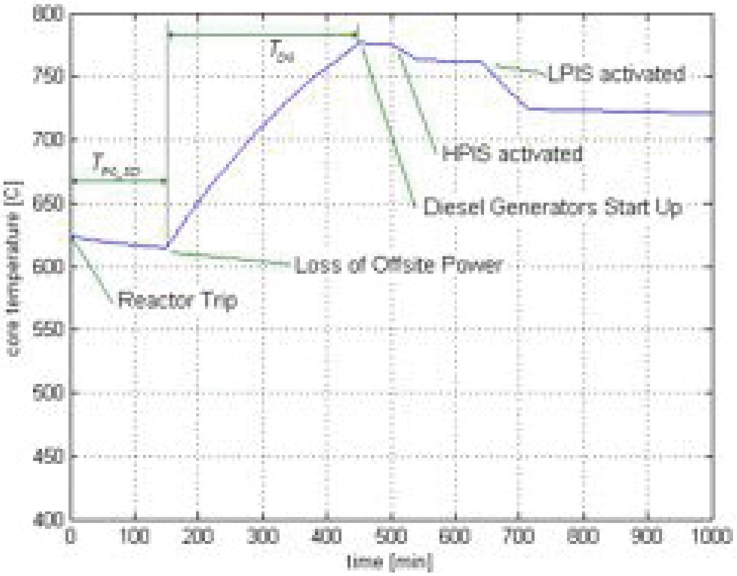


Figure 27: **NuclearF**: Example of a transient for a LOOP scenario.

is again evident that $\mathbf{M_1}$ and $\mathbf{M_2}$ based upon GPM introduce certain bias towards response surface fitting that more adaptively sampled points do not improve the fit drastically after 50 points or so. On the other hand, since $\mathbf{M_3}$ is not PM-dependent, it converges to the true limit surface (through visual inspection) reasonably fast after 30 adaptively sampled points.

4.3.2.4 RMSE Convergence

To further understand the convergence behavior of the limit surfaces estimated by $\mathbf{M_1}$ and $\mathbf{M_2}$, we obtain convergence plots in terms of the root mean squared error (RMSE) in Figure 29. All plots show the RMSE between estimated limit surfaces and the true limit surface[2], versus the number of samples added in the training data up to 100 points. The RMSE is computed using points located on the true limit surfaces.

It is important to note that $\mathbf{M_2}$ behaves comparably with $\mathbf{M_1}$ for general cases, e.g. for **MixtureA**, **MixtureB**, **MixtureC** and **DistanceD**; and for **DistanceD**, $\mathbf{M_2}$ converges to the same level a lot faster than $\mathbf{M_1}$. On the other hand, $\mathbf{M_2}$ appears to perform drastically better in certain situations, e.g. for **DistanceE** and **NuclearF**. The improved performance is due to the fact that, the topology-based model $\mathbf{M_2}$, with the tolerance parameter, allows exploration of potentially interesting regions of the domain that could contain failure regions. In the specific case of **DistanceE**, $\mathbf{M_2}$ manages to find the third failure region while $\mathbf{M_1}$ does not. In the case of **NuclearF**, $\mathbf{M_2}$ converges faster than $\mathbf{M_1}$. This is because during the early stage of adaptive sampling, $\mathbf{M_2}$ tries to explore the region topologically, placing a selected sampled point in the interior of the failure region, leading to faster convergence afterwards (as shown in Figure 28 second row).

To understand the average behavior of $\mathbf{M_1}$ and $\mathbf{M_2}$, in addition to RMSE plots with respect to a

[2] In the case where ground truth limit surface is not available, we could compute RMSE with respect to the previous round to estimate the convergence behavior.

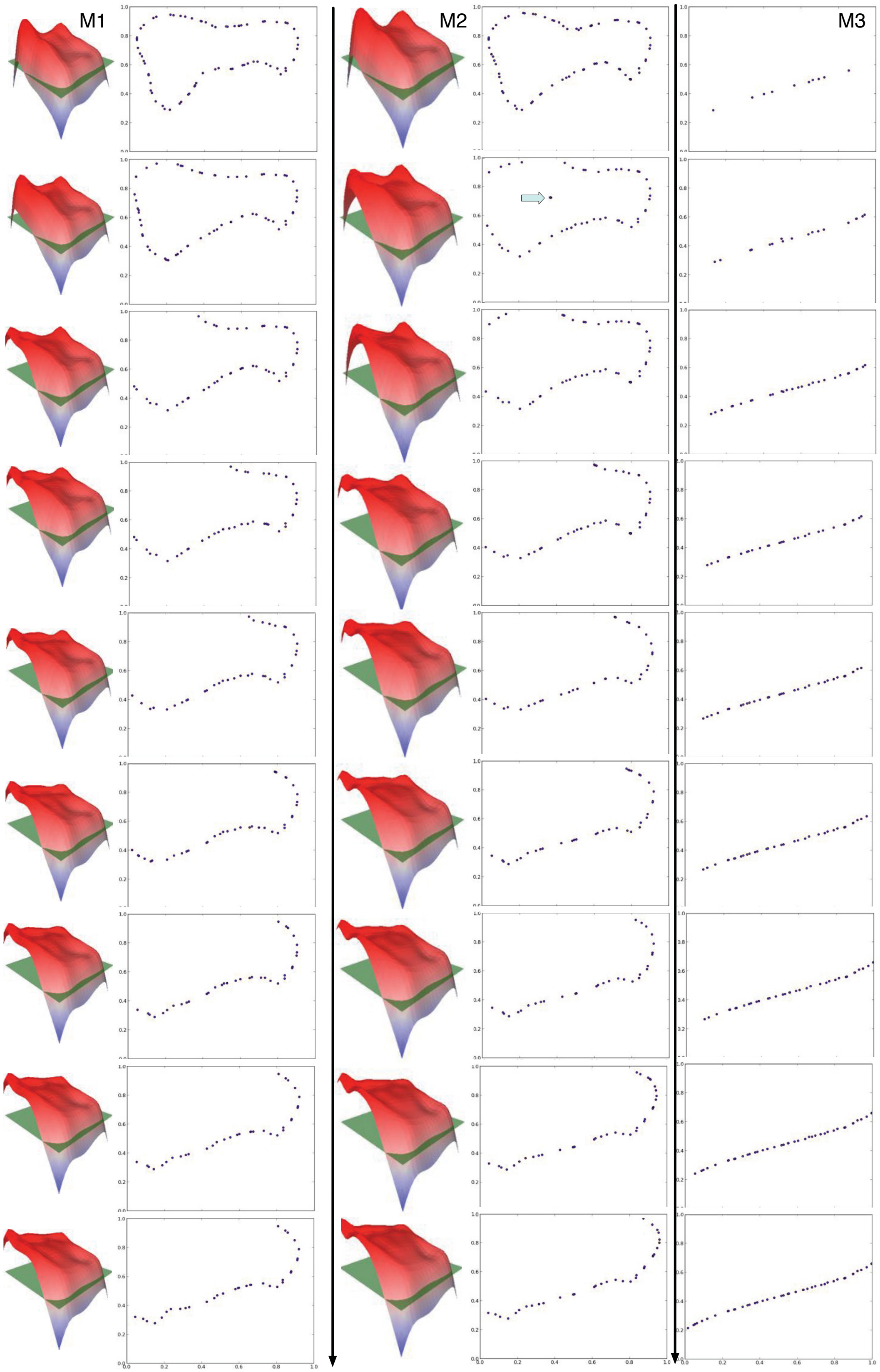


Figure 28: Convergence of estimated response surfaces and estimated limit surfaces for **NuclearF**, with all three models. Images are obtained after every additional 10 adaptively sampled points.

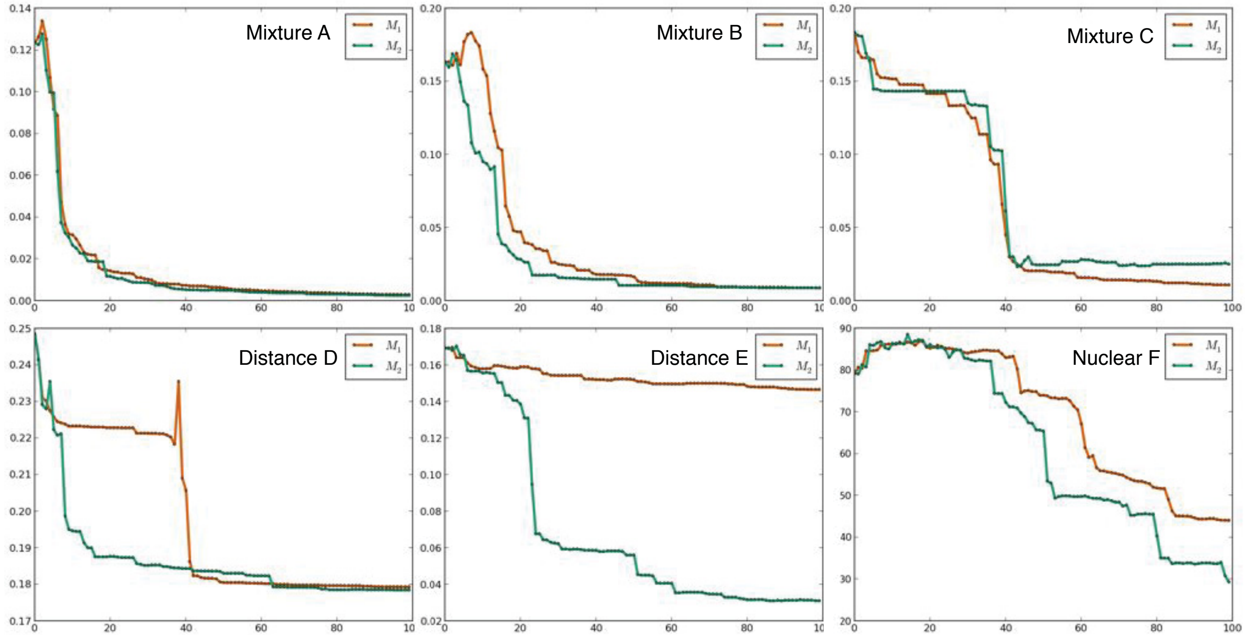


Figure 29: Convergence of RMSE of the estimated limit surfaces based on $\mathbf{M_1}$ and $\mathbf{M_2}$ for a single trial.

single trial, we compute RMSE with different initial training sets and obtain the mean and median plots across 10 trials for each dataset. The mean RMSE plots are shown in Figure 30, while the median RMSE plots are displayed in Figure 31. Our conclusions do not change much with respect to mean or median convergence behavior. In general, $\mathbf{M_1}$ and $\mathbf{M_2}$ perform comparably for majority of the datasets. $\mathbf{M_1}$ converges to a slightly lower RMSE than $\mathbf{M_2}$ for **MixtureC**, while $\mathbf{M_2}$ has a slight upper hand in the case of **DistanceD**. Both $\mathbf{M_1}$ and $\mathbf{M_2}$ converge to the same RMSE for **NuclearF**, however $\mathbf{M_2}$ converges slightly faster. In the case of **DistanceE**, $\mathbf{M_2}$ performs drastically better than $\mathbf{M_1}$ for both mean and median plots, which is expected as $\mathbf{M_2}$ recovers the third component in the limit surface which is missing from the other models.

Since $\mathbf{M_3}$ does not rely on a PM, therefore it is difficult to evaluate the estimated function values of the set of points located on the true limit surface from $\mathbf{M_3}$[3]. Thus we do not have RMSE convergence for $\mathbf{M_3}$. A more reasonable metric to measure the convergence of limit surfaces would be the dynamic time warping (DTW) distance, which is currently under consideration.

4.3.3 Discussions

In this section, we explore three adaptive sampling models to recover the limit surface. From $\mathbf{M_1}$ and $\mathbf{M_3}$, we learn a global model of the entire response surface using PMs such as GPMs or neighborhood graphs such as relaxed Gabriel graphs, and extract the limit surface as an iso-surface of the global model. For $\mathbf{M_2}$, we first study the topological segmentation of the global model and estimate

[3] However this is not entirely impossible, in 2D, this involves linearly interpolating within triangles; In high dimensions, this becomes a lot harder and requires construction of high-dimensional simplexes.

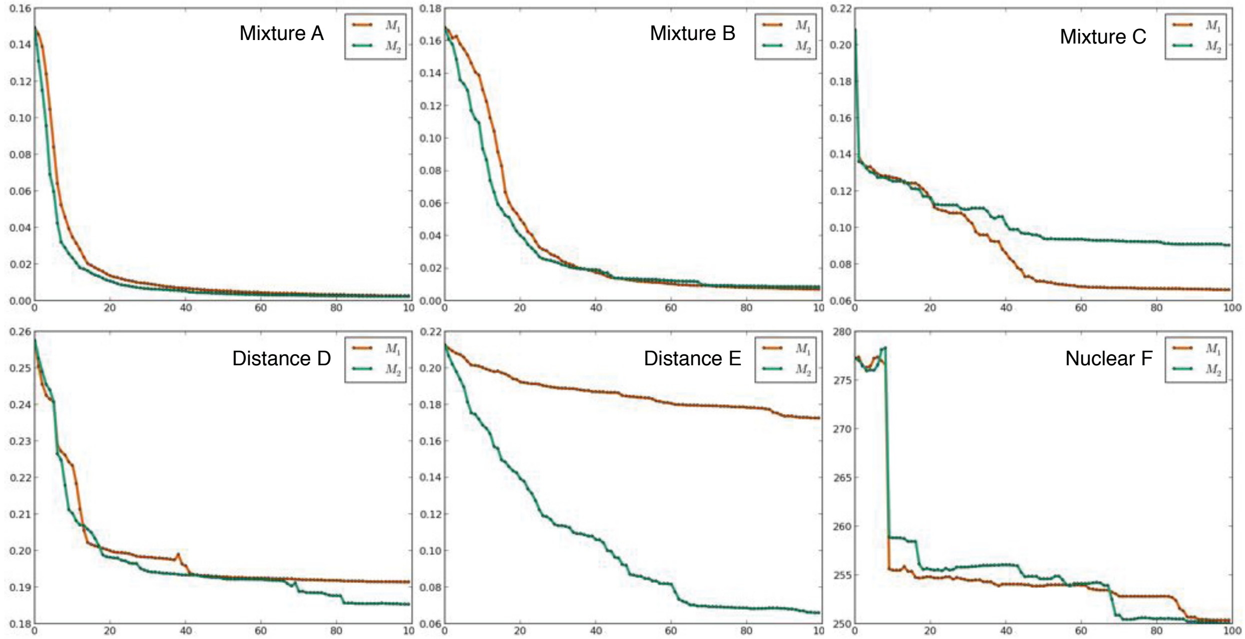


Figure 30: Convergence of mean RMSE of the estimated limit surfaces based on $\mathbf{M_1}$ and $\mathbf{M_2}$ across 10 trials.

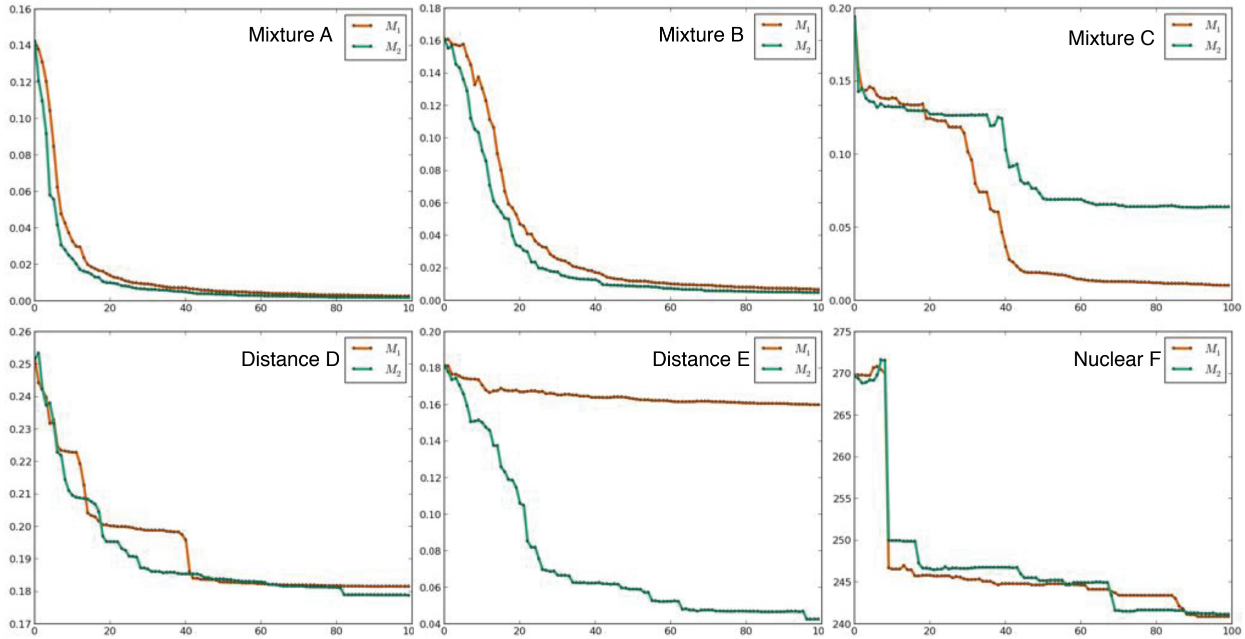


Figure 31: Convergence of median RMSE of the estimated limit surfaces based on $\mathbf{M_1}$ and $\mathbf{M_2}$ across 10 trials.

the limit surface locally through the relaxed spanning crystals of the corresponding Morse-Smale complex of the global model. We demonstrate our models on several analytical test functions as well as a nuclear simulation dataset modeled after a simplified PWR. All models typically perform well on most of the datasets tested in recovering the limit surfaces with reasonable accuracy. $\mathbf{M_2}$ performs comparably as $\mathbf{M_1}$ for most cases and under certain situations performs drastically better.

$\mathbf{M_1}$ and $\mathbf{M_2}$ fit the training data with a PM while $\mathbf{M_3}$ is completely data-driven and does not rely on a PM. $\mathbf{M_3}$ performs exceptionally well on the nuclear dataset as it does not suffer the bias from a fixed PM.

Among all three models, the topology-driven model $\mathbf{M_2}$ is most noticeable as it attempts to combine information from a coarse global representation using a PM with detailed local view based on topology. We expect $\mathbf{M_2}$ to have several advantages over some other adaptive sampling frameworks. First, $\mathbf{M_2}$ allows sampling densely around interesting regions of the response surface while avoiding uninteresting regions of the domain. In particular, it introduces a limit surface tolerance parameter δ that allows exploratory analysis of potential failure regions with controlled uncertainty. This phenomena could be investigated further in the near future where one could establish a multi-scale uncertainty characterization of the response surface by gradually varying δ. Second, we suspect that when dimension of the response surface is high and/or when the failure regions are small with respect to the entire domain area, $\mathbf{M_2}$ could be an efficient model which querys much smaller number of points from the PM. It is an on-going work to validate our claim by applying the model to datasets whose response surface is high dimensional and/or with small failure regions. We may further extend topology-driven adaptive sampling schemes to include other topological structures such as contour trees or Reeb graphs, which may offer new insights to our understanding of the response surfaces.

In summary, one would need to consider the tradeoff between exploration and exploitation when choosing the appropriate adaptive sampling model. A data-driven model such as $\mathbf{M_3}$ exploits the properties associated with existing training points with limited exploratory capabilities, while a PM-based model such as $\mathbf{M_1}$ focuses on sampling the entire domain. A topology-driven PM-based model such as $\mathbf{M_2}$ tries to strike a balance between the two through an uncertainty parameter δ. As δ increases, the model explores more uncertain neighborhoods while reduces samples in regions that strengthen the signal.

5 Topology-Based Clustering

5.1 Overview

Dynamic Probabilistic Risk Assessment (DPRA) [63] methodologies couple system simulator codes (e.g., RELAP [61], MELCOR [28], MAACS [12]) with simulation controller codes (e.g., ADAPT [34], ADS [38], MCDET [37], RAVEN [60]). While system simulator codes accurately model system dynamics (deterministically), simulation controller codes introduce both deterministic (e.g., system control logic, operating procedures) and stochastic (e.g., component failures, parameter uncertainties) elements into the simulation.

Typically, a DPRA is performed by 1) sampling values of a set of parameters from the uncertainty space of interest (using the simulation controller codes), and 2) simulating the system behavior for that specific set of parameter values (using the system simulator codes). For complex systems, the major challenges in using DPRA methodologies, especially those that employ dynamic event trees (DETs), are the heavier computational and memory requirements needed to search the uncertain parameter space systematically and to generate a large number of scenarios due to the time evolution of a large number of variables.

The analysis of such data is normally achieved by considering the end state of each simulation run (e.g., fuel melting temperature reached v.s. intact core) and observing the sequence and the timing of events that lead to that end state. When system complexity is very high (e.g., analysis of nuclear power plant accident scenarios), analysis of such large quantities of data may require advanced tools that evaluate impact of uncertainties and timing/sequence of events on system dynamics.

A first approach toward discovering these correlations from data generated by DPRA methodologies has been developed using Fuzzy classification [47] and clustering algorithms [54]. In particular, clustering algorithms partition the set of scenarios into clusters by identifying similarities based on certain criteria, allowing users to organize and interpret the trends in scenario evolution and risk contributors for each initial event [67]. The set of scenarios could be analyzed in two modes, either by *end state analysis* that classifies the scenarios into clusters based on their end state (e.g. final outcome) (e.g. [68]), or by *transient analysis* that considers the complete system dynamics (e.g. time evolution of scenarios) and identifies clusters having similar temporal behavior of the state variables (e.g. [54]). The work that is of most relevance to us utilizes a mode-seeking clustering method, the mean-shift algorithm [54]. The mean shift algorithm [25] is a non-parametric iterative procedure that assigns each data point to the average of data points in its neighborhood as the cluster center. The data points are therefore clustered based on observation density.

In this section, we present a software tool that provides the domain experts with an interactive analysis and environment for understanding the structures of high-dimensional nuclear simulation datasets. In particular, such a tool enables the end users to perform both *end state analysis* that apply to the end state of the scenarios, and *transient analysis* that apply to their time evolution data, to classify the set of nuclear scenarios. We focus on clustering algorithms based on topological structures such as the Morse-Smale complex, which partitions the data points into clusters based on their uniform gradient flow behavior. We compare our techniques with mean shift methodology on a dataset generated by a DET for a sodium-cooled fast reactor during an aircraft crashing scenario. We demonstrate that these techniques offer complementary views of the data that help illuminate key features that may be otherwise hidden using a single methodology.

5.2 Analysis Methodology

We apply clustering algorithms based on Morse-Smale complex to nuclear datasets generated by DETs. We model such a dataset as a high-dimensional scalar function f defined on a finite set of points $\mathbb{X}$ in $\mathbb{R}^n$. We partition the points in $\mathbb{X}$ based on their function values and gradient behavior with respect to the approximated Morse-Smale complex. That is, points belong to the same cluster if they have uniform gradient flow behavior.

5.3 A RVACS Case Study

Demo example. To illustrate the proposed clustering methodology, we use a dataset generated for the analysis of recovery from an aircraft crash into an RVACS of a conceptual design for a sodium-cooled fast reactor [54,55,65,66]. The RVACS is a passive decay-heat removal system that removes heat by natural circulation of air in the gap between the vessel and a duct surrounding the vessel. With this system, the reactor decay heat is released to the atmosphere through four cooling towers. The Analysis of Dynamic Accident Progression Trees (ADAPT) tool [34] is used as the DET generator while the system dynamics is modeled using RELAP5 [61]. A typical scenario is the following: the plant is operating at 100% power when an airplane crashes into the plant, destroying three of the four towers and, thus, the reactor core cooling capabilities are disabled. A recovery crew then arrives at the site and attempts to reestablish the cooling of the reactor by restoring the damaged towers one by one. An ensemble of 609 transient simulations has been generated[4], and among which 132 scenarios are considered system failures when the reactor reaches a maximum temperature of 1000K before the end of simulation, which is fixed to be 2×10^5 seconds[5]. The rest of the 477 scenarios are considered simulation completions. Each simulation includes information regarding: (a) Time profile of core temperature; (b) Crew arrival time; and (c) Recovery time of tower 1, 2 and 3.

Data representation for end state analysis. For the end state analysis, the above data is represented as various 4D scalar functions. We employ two types of representations: 4D scalar function based on absolute timing and relative timing, respectively.

For the 4D absolute timing analysis, the four input parameters represent the time for the crew to arrive at the plant (t_0), and the time for them to recover the first (t_1), second (t_2) and third (t_3)

[4] One transient simulation out of 610 simulations is considered as an outlier due to discrepancies among crew arrival times.

[5] Scenarios that reach 1000K before the end of simulation have been extended in time with the last value simulated.

Label	Description	Label	Description	Label	Description
A.1	4D-MT-all-2C	B.1	4D-Delta-MT-all-5C	C.1	100D-MT-all-4C
A.2	4D-MT-completions-4C	B.2	4D-Delta-MT-completions-5C	C.2	100D-MT-completions-3C
A.3	4D-TTF-all-2C	B.3	4D-Delta-TTF-all-4C	C.3	100D-TTF-all-3C
A.4	4D-TTF-failures-3C	B.4	4D-Delta-TTF-failures-4C	C.4	100D-TTF-failures-3C
A.5	4D-TTM-all-3C	B.5	4D-Delta-TTM-all-4C	C.5	100D-TTM-all-3C
A.6	4D-TTM-completions-3C	B.6	4D-Delta-TTM-completions-4C	C.6	100D-TTM-completions-3C

Table 1: Various 4D and 100D functions under analysis. A: End state absolute timing analysis. B: End state relative timing analysis. C: Transient analysis.

tower, respectively. In certain cases, the third tower has not been recovered before the end of the simulation, so $t_3 = 0$. The output variables considered are: (1) MT: maximum temperature reached in the simulation; (2) TTF: time to reach failure temperature[6]; (3) TTM: time to reach maximum temperature. In other words, in the Morse-Smale complex formulation, the finite set of points $\mathbb{X}$ is embedded in $\mathbb{R}^4$ with coordinates coming from the 4 input parameters, and the scalar function f defined on $\mathbb{X}$ obtains its values as one of the three figures of merit (e.g. MT, TTF or TTM).

For the 4D relative timing analysis, the four input parameters represent the relative time between recovering procedures, that is, the time for the crew to arrive at the plant from the time when the accident occurs ($t'_0 = t_0 - 0 = t_0$), the time between recovering the first, second and third tower respectively, that is, $t'_1 = t_1 - t_0$, $t'_2 = t_2 - t_1$, $t'_3 = t_3 - t_2$. The output variable of interest is designed to be MT, TTF or TTM.

Data representation for transient analysis. For the transient analysis, since only a single state variable (e.g. the core temperature) is under consideration, we model the temporal evolution of each scenario as a d-dimensional point, where d is the number of uniform temperature samples obtained along its time profile. We set $d = 100$. Therefore, the above data is modeled as various 100D scalar functions. Again, the output variable of interest is designed to be MT, TTF or TTM.

Cases under analysis. A further analysis of these various 4D and 100D functions based on all scenarios, system failure scenarios and simulation completion scenarios leads to the detailed clustering results for the following cases summarized in Table 1. The shorthand description of each case encodes the dimension, output variable of interest, scenarios included and number of crystals obtained. For example, case *(A.1) 4D-MT-all-2C* means a 4D function under the end state absolute timing analysis, where the output variable is MT, and the obtained result includes 2 crystals. Similarly, case *(B.4) 4D-Delta-TTF-failures-4C* means a 4D function under the end state relative timing analysis for the system failure cases, where the output variable is TTF and the result includes 3 crystals; case *(C.6) 100D-TTM-completions-3C* means a 100D function for the simulation completion cases under the 4 crystal setting, where the output variable of interest is TTM.

Objective. We would like to understand how these input variables impact system dynamics [55, 66]. Understanding the structure of such 4D and 100D functions may help domain scientists to make decisions regarding repair strategies and evacuation plans.

[6] For scenarios where failure temperature is never reached, this is set to 2×10^5.

5.3.1 Visual Interface

We review and illustrate our visual interface in Figure 32 through the analysis of a 4D function under case (A.1) described above. Such visual interface is inherited and extended from the capabilities provided by HDViz [29] and has been previously employed for nuclear reactor analysis and visualization [50]. As shown in Figure 32 (a), the topological summary interface summarizes each Morse-Smale crystal into a 1D curve in high-dimensional space which is then projected onto a viewable 3D space. The interface encodes three steps (detailed in [29]), to arrive at a 3D representation for analysis and visualization of the d-dimensional scalar function f, defined on a set of sampled points $\mathbb{X}$. (1) *Morse-Smale approximation*: We approximate the Morse-Smale crystals in high dimension using a relaxed Gabriel graph. (2) *Geometric summaries:* For each crystal of the Morse-Smale complex, since each point has similar approximated monotonic gradient behavior, a geometric summary is constructed by an inverse regression, yielding a 1D curve in the d-dimensional domain of f. Intuitively, the value of the parametrized curve at a given location yields a representation of the crystal as the average of the function values of level sets within the crystal. (3) *Dimension reduction:* The set of regression curves can be represented by a graph embedded in $\mathbb{R}^d$ with each edge corresponding to a curve and vertices corresponding to extremal points. Such a graph is then embedded into 2D preserving the spatial relation among the extrema and the geometry of the crystals that connect them using PCA or ISOMAP [64], while the third dimension is reserved for the output parameter. A simple example of the above three step process is shown for a simple 2D height function in Figure 33.

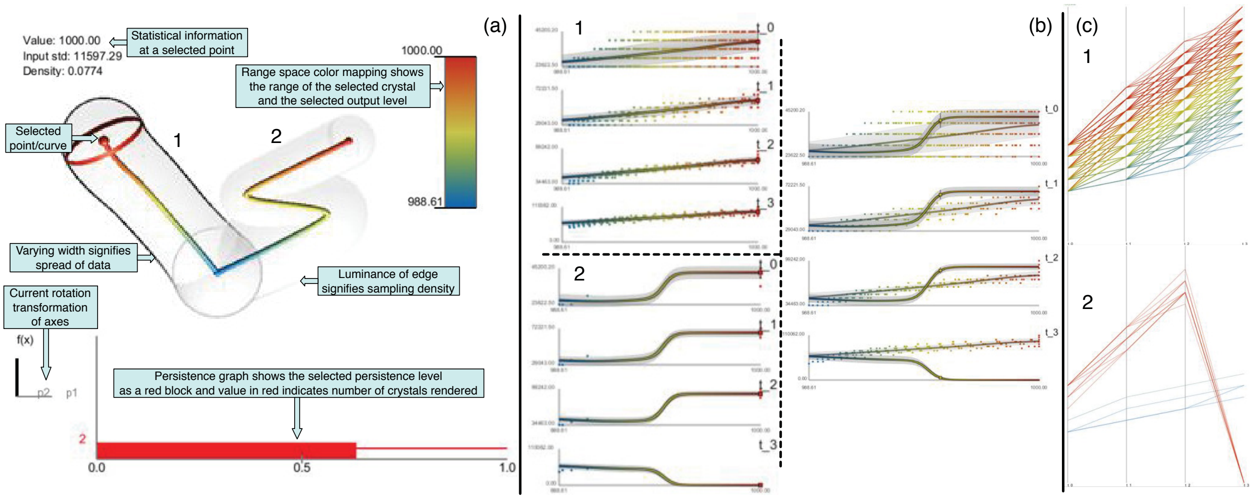


Figure 32: Visual interface illustrated through analysis of 4D function described in (A.1). (a) The topological summary visual interface. (b) Left: Inverse coordinate plots for crystals 1 and 2 in (a); (b) Right: combined inverse coordinate plots for crystals sharing the same minima; (c) Parallel coordinate plots for crystals 1 and 2 in (a).

Various visual components are described in Figure 32 (a), three of which are of most relevance to us in the current context: the crystal projection, the persistence graph and the range space color mapping. To enable multi-scale analysis, we use a modified version of the persistence diagram [24], referred to as the *persistence graph*. It shows the number of Morse-Smale crystals

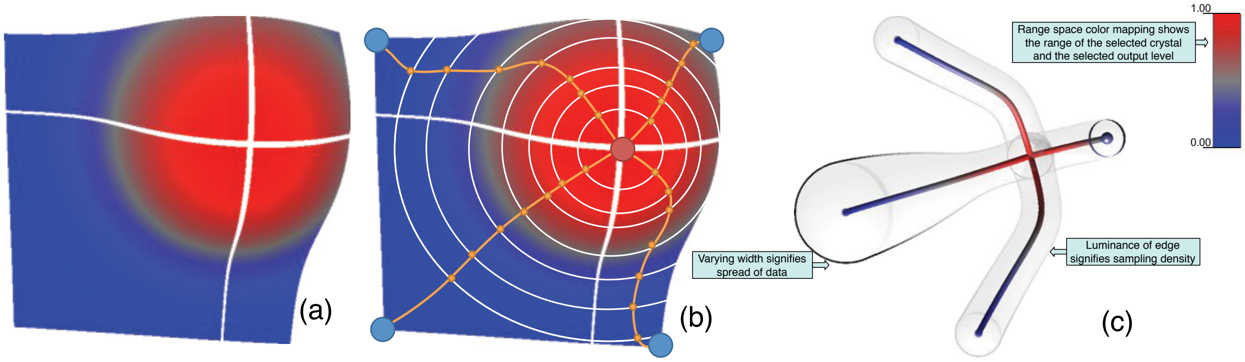

Range space color mapping shows the range of the selected crystal and the selected output level
1.00
0.00
Varying width signifies spread of data
Luminance of edge signifies sampling density
(a)
(b)
(c)

Figure 33: Key steps in generating the topological summary of a simple 2D height function. (a) Morse-Smale crystal approximations that decompose the domain into 4 crystals. (b) Each crystal obtains a 1D geometric summary which encodes the average of function values at level sets within the crystal. (c) The collection of 1D curves are then projected to a viewable 3D space preserving the spatial relations among the extrema and the geometry of the crystals. Several important visual components are described here as well.

(y-axis) as a function of scale (i.e. x-axis, persistence threshold normalized by the range of the dataset). A selected scale is drawn with a red box and a corresponding number of Morse-Smale crystals is displayed along the y-axis in red. Stable features are considered as those that exist over a large range of scales (i.e. a sequence of persistence simplification with increasing scales), which correspond to long horizontal lines in the persistence graph.

In the inverse coordinate plots, each input parameter (e.g. t_0) is considered as a 1D function of the output variable (e.g. MT). This is shown in Figure 32(b), where the 1D regression curve as well as the data points are colored according to range space values. Furthermore, we could create combined inverse coordinate plots by superimposing those from crystals that belong to the same ascending/descending manifold. This enables further differentiations among various dimensions across crystals. We further use parallel coordinate plots [39] to illustrate the correlations among input parameters in the high dimensional datasets for each crystal, as shown in Figure 32 (c), where the curves are colored based on range space values as well.

To highlight clustering structures obtained from the topological segmentation, we change our color map of the above visual interface where the 1D regression curve as well as data points are colored based on clustering membership, as shown in Figure 34.

5.3.2 End State Analysis

For this section, we showcase some insights obtained from various analysis cases. These insights and their implied conclusions are only a fraction of possible outcomes from our analysis and visualization framework, and a lot of these observations demand further investigations. Nevertheless, they give us some initial understanding of the nuclear dataset analyzed, and offer complementary views compared to results obtained with mean-shift methodology [54]. For each of the following

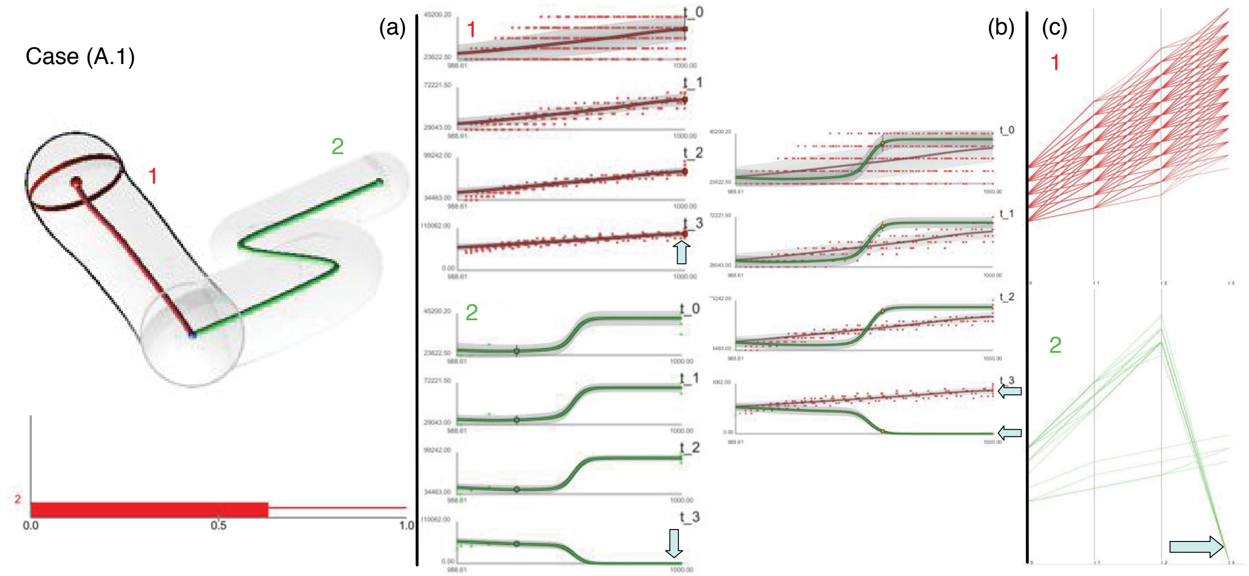


Figure 34: Visual interface highlighting clustering structure for 4D function described in (A.1). (a) The topological summary visual interface. (b) Inverse coordinate plots for both crystals individually and combined. (c) Parallel coordinate plots.

cases, the number of crystals (i.e. number of clusters) are chosen based on the persistence graph. In the situations where several possible clusterings with different cluster numbers give comparable results, we choose one of the clusterings arbitrarily. We use the word cluster and crystal to mean the same thing. Important observations/insights are obtained from the part of the visual interface highlighted by arrows in cyan.

4D Absolute Timing Analysis. The results are shown in Figure 34 for case (A.1) and Figure 35 for cases (A.2) through (A.4) and Figure 36 for cases (A.5) and (A.6). We now describe each case in detail. For **(A.1) 4D-MT-all-2C**, both crystals have maximum temperature of 1000K. Crystal 1 (red) contains scenarios where $t_3 \neq 0$, however, in these scenarios, the maximum temperature reaches the failure temperature of 1000K. This means that even though the crew managed to recover all three towers, since the recovery time for all towers is high, the core temperature keeps rising reaching system failure anyway. Crystal 2 (green) contains all scenarios where $t_3 = 0$, indicating all scenarios where the 3rd tower is not recovered and the failure temperature is reached. For **(A.2) 4D-MT-completions-4C**, we only analyze simulation completion cases. The four local maximum have temperatures 999.98K, 999.79K, 999.80K and 999.61K respectively. Crystal 1 (green) and 2 (red) have comparable range of values for t_0, t_1 and t_2, however crystal 1 contains data points with higher t_3 values compared to crystal 2, this indicates a later recover time for tower 3. Or in other words, recovery time for tower 3 differentiates these two crystals. Crystal 3 (blue) and 4 (purple) contain scenarios characterized by small values of t_0 and high values of t_1, t_2 and t_3, i.e., early crew arrival time and late recovery of the 3 towers. When looking at the combined inverse regression plots based on the ascending manifold, we see that the crystals 1 and 2 contain data points with higher t_0, compared to the other two crystals, indicating late crew arrival time. For **(A.3) 4D-TTF-all-2C**, crystal 2 (green) contains all cases where $t_3 = 0$. For **(A.4) 4D-TTF-failures-3C**, we exclude non-failure scenarios. The two local maxima correspond

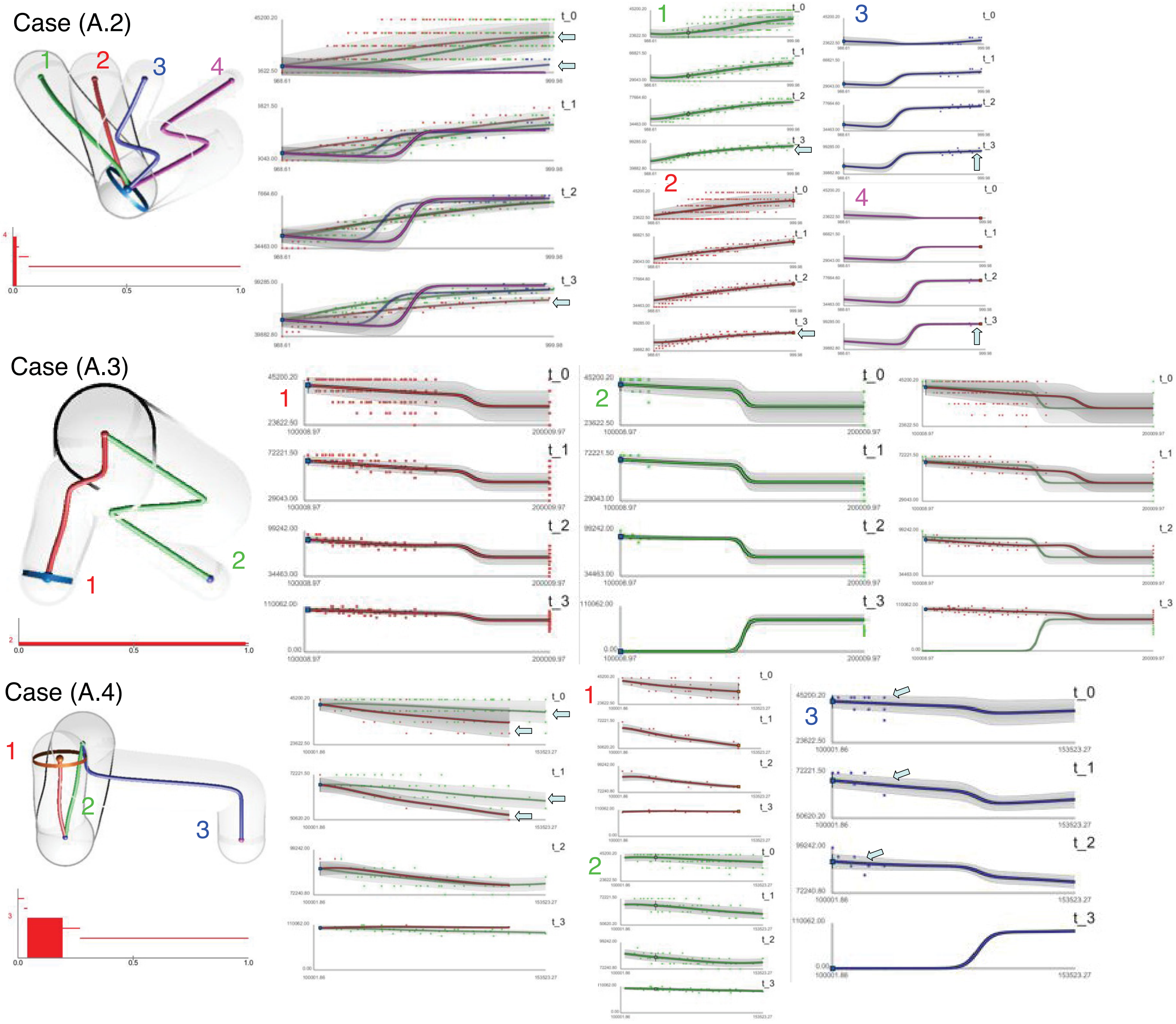


Figure 35: End state analysis, cases (A.2) through (A.4).

to approximately 1.45×10^5 seconds and 1.54×10^5 seconds. The two local minima are roughly 1.01×10^5 seconds and 1.00×10^5 seconds. Crystals 1 (red) and 2 (green) both contain significant number of scenarios. However, they do not share the same local maxima, that is, time to reach failure in their local regions. Based on the combined inverse coordinates plot of crystals on the ascending manifold, they can also be differentiated by variations in the general trends of t_0 and t_1. In particular, crystal 3 (blue) contains scenarios that lead to an early time to reach failure characterized by late crew arrival and late recovery of the first 2 towers. In **(A.5) 4D-TTM-all-3C**, the three local minimum corresponding to roughly 1.00×10^5, 1.01×10^5 and 1.05×10^5 seconds. Crystal 1 (green) contains all scenarios where $t_3 = 0$, and some scenarios where the crew arrived at the site early (where t_0 is small) and the temperature peaked (reached its maximum) late. Both

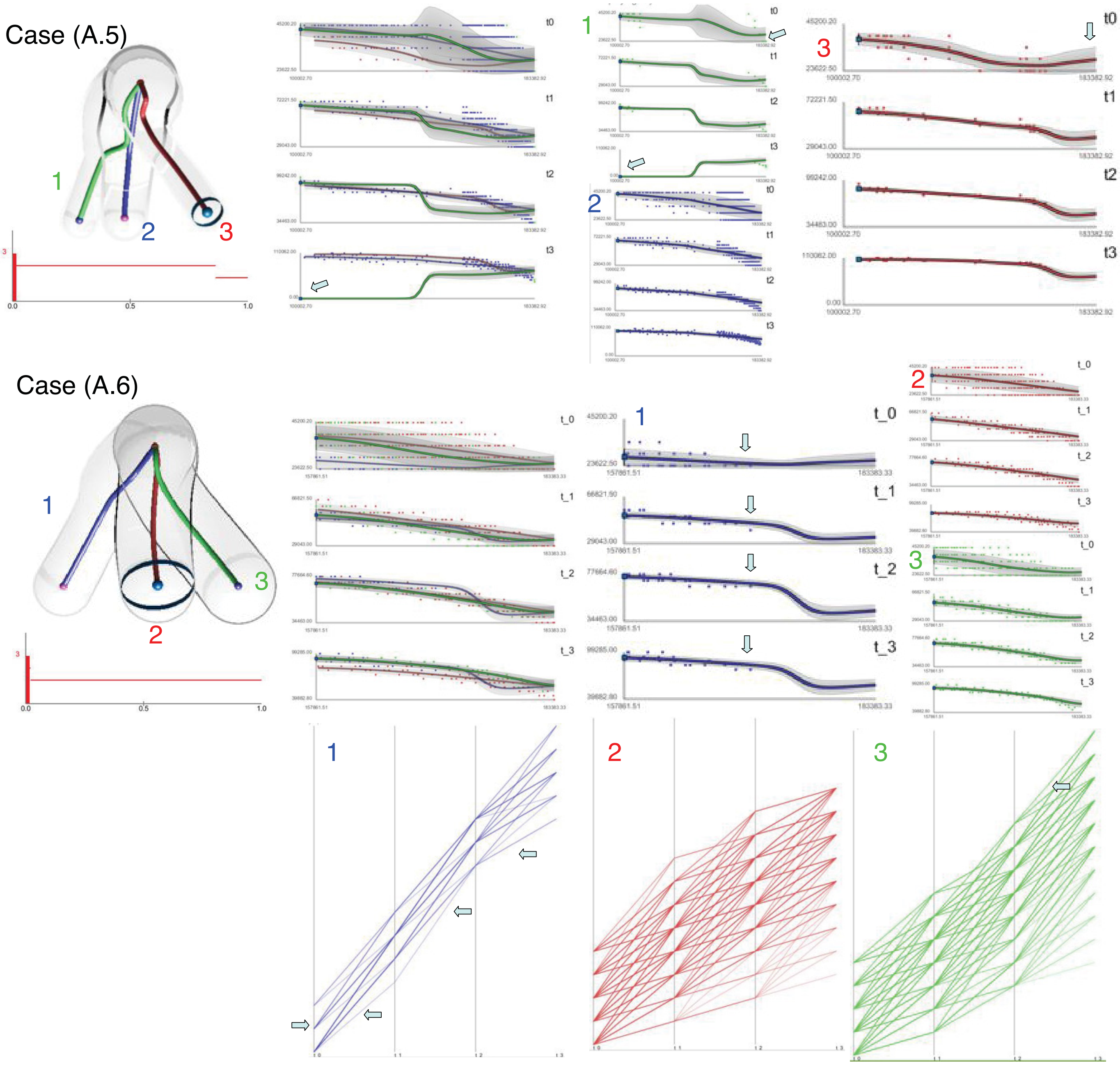


Figure 36: End state analysis, cases (A.5) and (A.6).

crystals 1 and 2 (blue) have some scenarios of the crew arriving early and the temperature peaking late, while crystal 3 (red) has none of the extreme cases. Finally, in **(A.6) 4D-TTM-completions-3C**, the three local minimum are very close in value, roughly around 1.58×10^5 seconds. Crystal 1 (blue) contains purely scenarios that reach maximum temperature early, with early crew arrival time and a large gap between time to fix the towers (see the large slope in its corresponding parallel coordinate plot). Between crystal 1 and crystal 3 (green), v.s. crystal 2 (red), the former two have higher gap between t_2 and t_3 (large slope in the parallel coordinate plots), meaning a big delay in recovering tower 3. Again, late crew arrival and late tower recovery times cause these scenarios to reach the maximum temperature early; while early crew arrival and early tower recovery times result in reaching maximum temperature late.

4D Relative Timing Analysis. The results are shown in Figure 37 with detailed descriptions below. We omit cases (B.2), (B.4) and (B.6) since they do not lead to drastically different conclusions. For **(B.1) 4D-Delta-MT-all-5C**, the two local maxima correspond to 991.66K and 1000K. The four local minima are valued at 990.48K, 990.5K, 988.61K and 990.15 respectively. Crystal 1(orange) contains only a single data point. However it is persistent across multiple scales up to two crystals. It also reaches the lower local maxima temperature of 991.66K, and corresponds to the largest t'_3 value (according to the parallel coordinate plot), indicating a very late recovering time for tower 3. Both crystal 3 (purple) and 4 (green) contain some cases where $t'_3 = 0$. Crystal 3 contains late crew arrival (e.g. large t'_0) scenarios and some early tower three fixing time (small t'_3). Crystal 2 (red) contains late t'_3 while crystal 5 (blue) contains late t'_2 and moderate t'_3 scenarios. In **(B.3) 4D-Delta-TTF-all-4C** and **(B.5) 4D-Delta-TTM-all-4C**, both corresponding crystal 1 (blue) contain all cases where $t'_3 = 0$ that is $t_3 = 0$, meaning no recovering of tower 3. These two cases have extremely similar clustering structure, which may be interesting to further investigate.

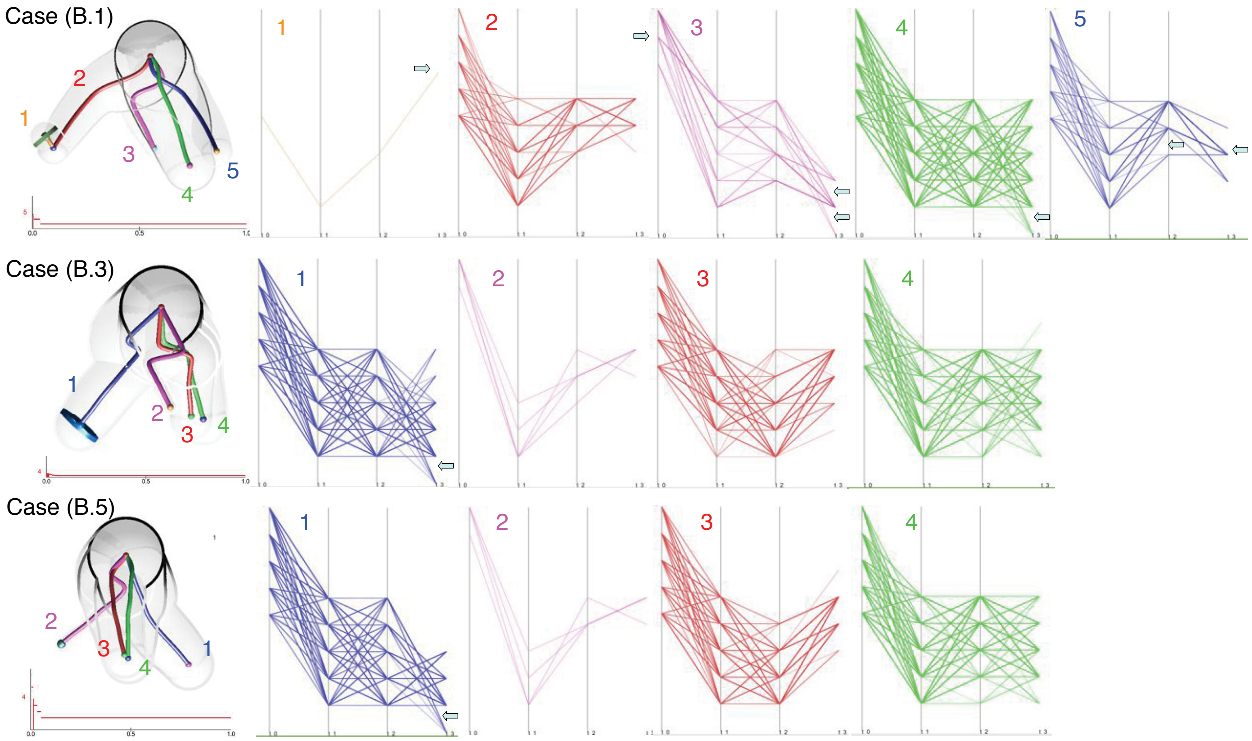


Figure 37: End state analysis, cases (B.1), (B.3) and (B.5).

5.3.3 Transient Analysis

For transient analysis results, we first focus on case **(C.1) 100D-MT-all-4C**, illustrated in Figure 38. As shown in (a), the analysis results in 4 clusters, and the 4th cluster contains most of the

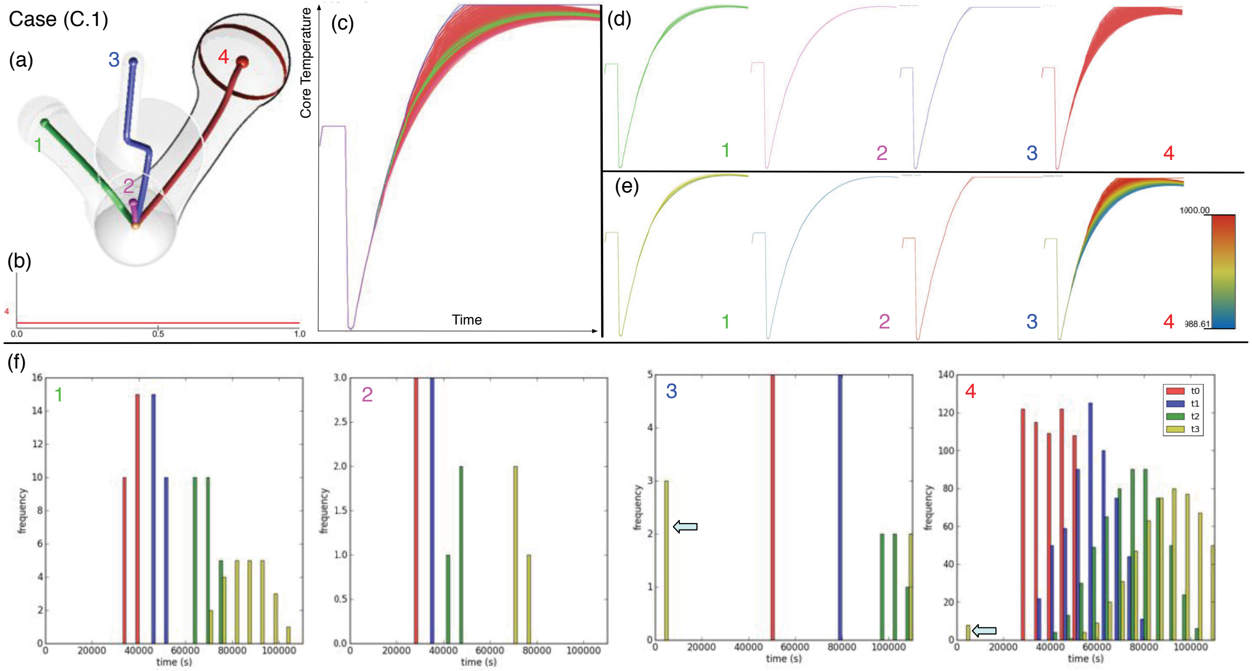


Figure 38: Transient analysis for case (C.1). (a) Topological summary with 4 crystals, corresponding to 4 clusters colored accordingly. (b) Persistence graph under the 4 crystal setting. (c) Parallel coordinate plot combining all 4 clusters, colored by cluster labels, showcasing time-varying profiles of all scenarios: x-axis corresponds to 100 values sampled along time, y-axis indicates core temperature. Parallel coordinate plots corresponding to each individual cluster, colored by cluster labels in (d) and by function values in (e). (f) Distribution of crew arrival time t_0 (red) and the recovery time of tower 1 (blue), 2 (green) and 3 (yellow) for each cluster.

scenarios in (d). As for MT profiles in (e), cluster 2 contains scenarios with the lowest MT, while cluster 3 contains some of those with the highest MT. For each cluster, the distributions of crew arrival time and the recover time of each tower are shown in (f). Note that for cluster 3 and 4, part of the distribution for t_3 is concentrated around 0 which correspond to the failure scenarios where the crew was not able to recover tower 3 on time (e.g. $t_3 = 0$). Cluster 3 is characterized by a fixed range of crew arrival time (t_0) and an extreme late crew recover time of tower 3 (t_3), this explains the high MT obtained by thoes scenarios.

Transient analysis results for (C.2) and (C.3) are shown in Figure 39. For **(C.2) 100D-MT-completion-3C**, among all completion scenarios, cluster 2 contains those with the lowest MT, which is also characterized by very early crew arrival time and relatively early tower 3 recover time, leading to low MT. On the contrary, cluster 1 has late crew arrival time and late tower 3 recover time, leading to scenarios with some of the highest MT. In **(C.3) 100D-TTF-all-2C**, cluster 3 contains those with the highest TTF, while cluster 1 contains those with the lowest TTF. Scenarios in cluster 1 reach failure temperature earliest because they are characterized by late crew arrival time and late recovering time for all towers.

For completeness, the remaining transient analysis results are shown in Figure 40. As we could

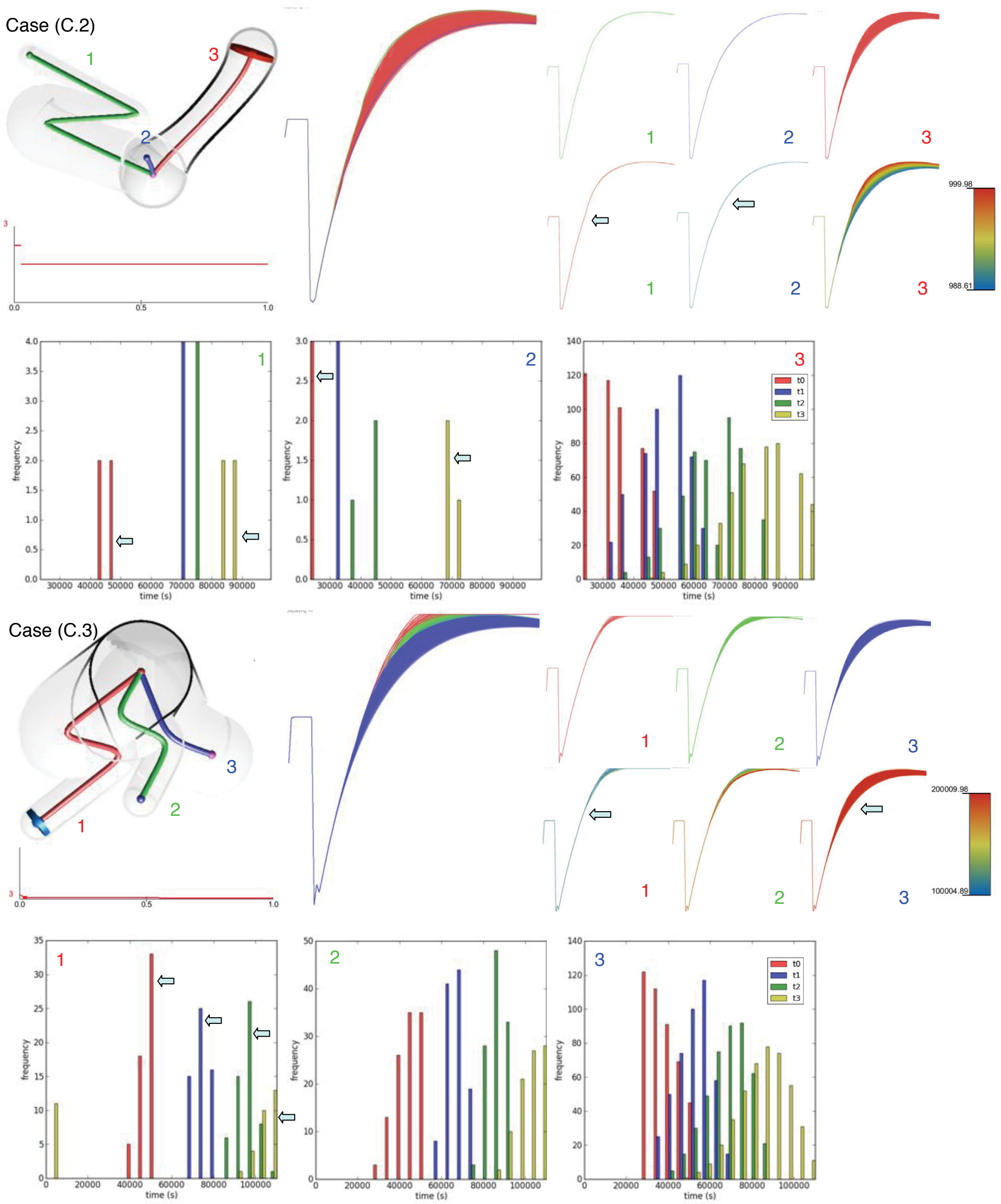


Figure 39: Transient analysis, cases (C.2) and (C.3).

observe from these figures, for cases (C.4), (C.5) and (C.6), there are clusters with distinctive patterns for the distributions of the variables t_0, t_1, t_2 and t_3, differentiating them from one another.

Complementary Views. Now we could combine the above transient analysis results based on topological clustering, with previous results obtained through mean shift methodology ([54], Figure 19 and Figure 20). We argue that both methodologies offer complementary views of the nuclear simulation dataset. Further investigations are currently underway to understand how such views help the end users in the drawing of important conclusions with respect to safety analysis.

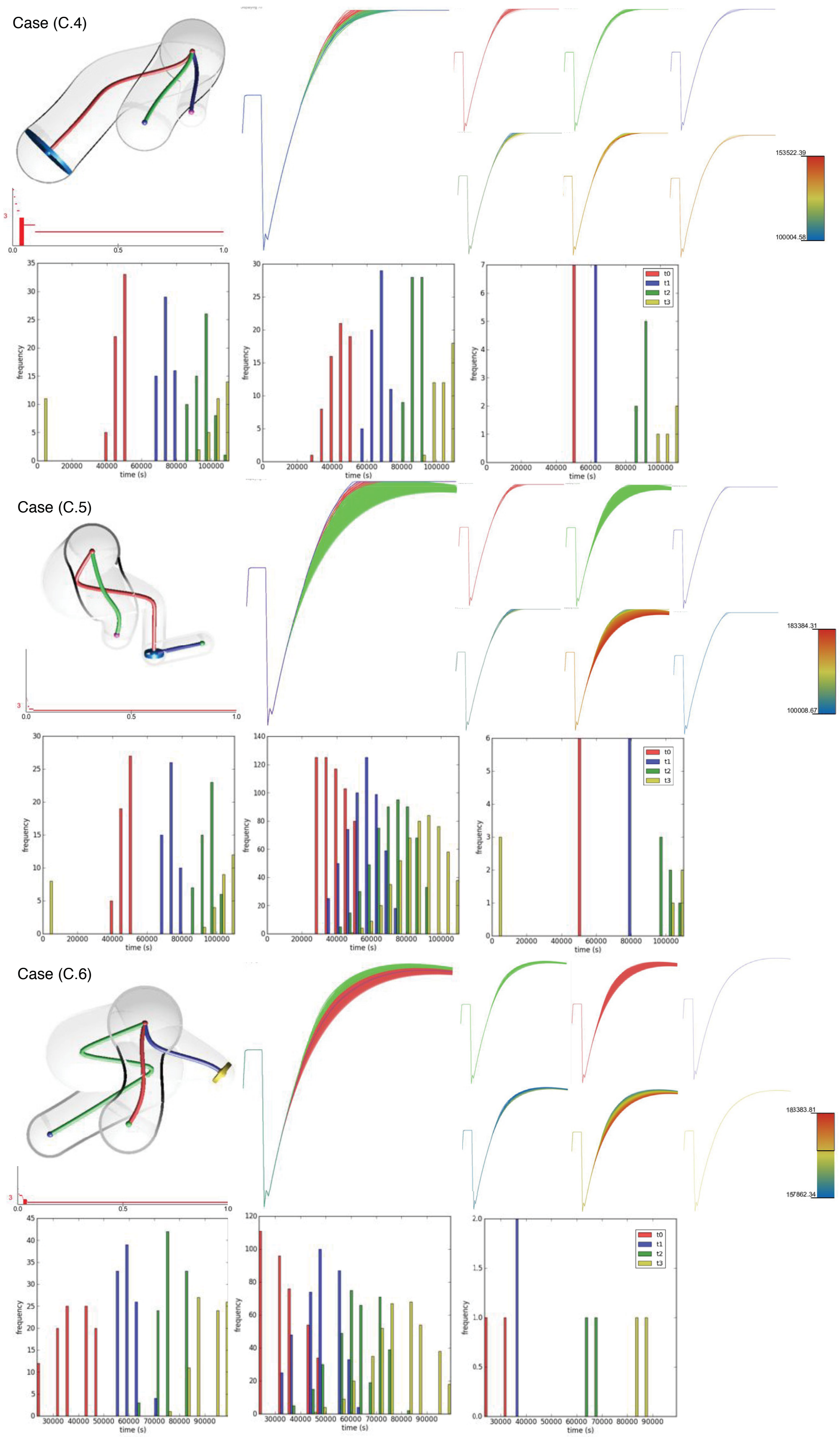


Figure 40: Transient analysis, cases (C.4) to (C.6).

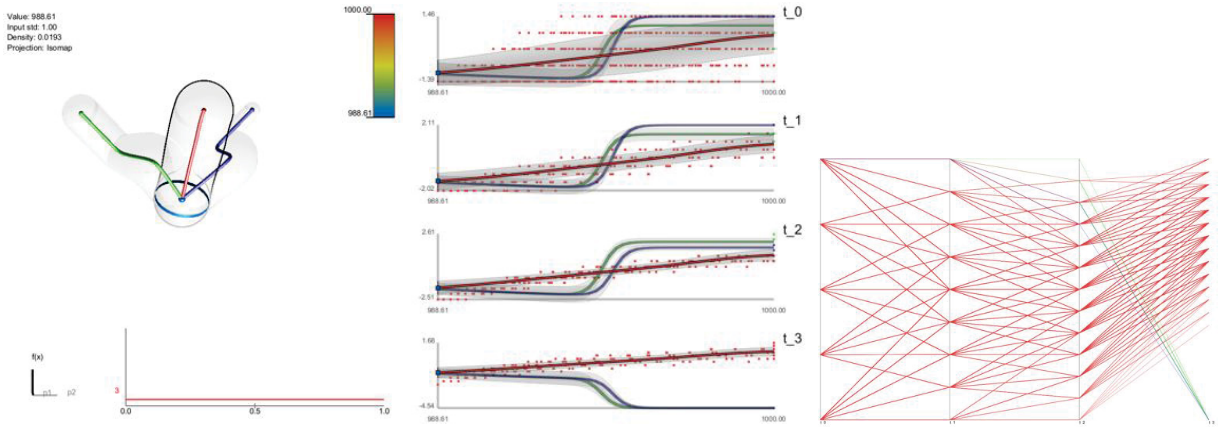


Figure 41: **4D-MT-All-3C**: topological summary (left), inverse coordinate plots with (middle) and without (right) points projection. Points and regression curves are colored by cluster (crystal) memberships.

5.3.4 Analysis and Visualization with Data Standardization

All the 4D cases described previously are based upon the raw data points without standardization. As discussed in Section 3, non-standardized and standardized data may have very different topological structures depending on how each dimension is weighted in computing the Euclidean distances. This has been studied in detail in [56]. In this section, we present the different analysis and visualization results when the 4D data is standardized based on the Z-score scaling.

4D-MT-All-3C. For visualization, we have selected a point in the topological hierarchy where there are three topological crystals that share a single, global minimum and diverge into three, distinct local maxima. The shared minimum simulation is characterized by a maximum temperature of 988.61, whereas each of the three maxima represent failure simulations where the maximum temperature reached was recorded as 1000 before the simulation was halted. The blue and green crystals in this set of simulations compose all of the cases where the simulation resulted in failure before the third tower was able to be fixed. This is encoded as a t_3 value of 0 which we can see as all of the green and blue simulations in the parallel coordinate plot in the right image of Figure 41 plummet on the last axis. All of the success cases and failure cases that occurred after fixing the third tower are incorporated into the red crystal. The blue crystal contains the latest values of t_0, but so does the red crystal. More notably, the green and blue crystals have t_1 and t_2 values that are as late or later than all of the red crystal simulations. The blue crystal contains points with the lastest t_1 values, but the green crystal contains the simulations with the latest t_2 values, which can be used to distinguish these separate failure cases from one another.

4D-MT-Delta-All-4C. In this analysis, we take each of the relative timings for each of t_0, t_1, t_2, and t_3, that is t_1 is the time from each simulations t_0 to fix tower 1, not the time from $t = 0$, and standardize each dimension to have zero mean and a standard deviation of one. We visualize the results in Figure 42, and note there is a large local maximum that attains a value of 1000, indicating system failure and core damage, and a smaller local maxima that attains a maximum temperature of

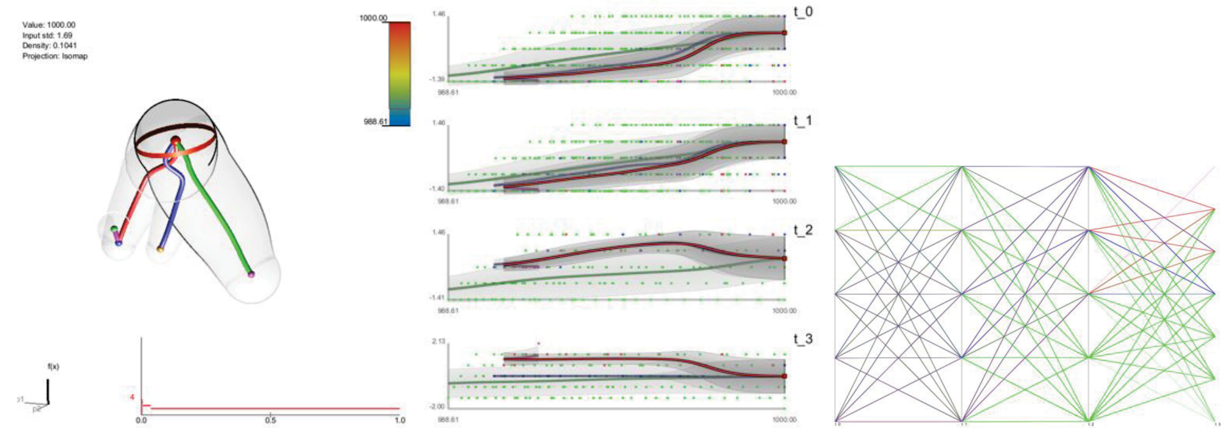


Figure 42: **4D-PCT-All-5C**: topological summary (left), inverse coordinate plots with (middle) and without (right) points projection. Points and regression curves are colored by cluster (crystal) memberships.

991.66, thus all scenarios in the associated magenta crystal represent success scenarios. The local minimum shared between the two maxima has a temperature of 990.48, thus the magenta curve represents a very small coverage of the range space. The remaining blue and green crystals diverge to their own local minima that have values of 990.15 and 988.61, respectively. One can note the variance in minimum settings is rather low, but threre are significant differences in the domain space of each of these occurrences that leads to them being separated into different crystals. The blue and red crystals have the same setting in the first three dimensions (t_0,t_1,t_2), but the red crystal has a significantly later time to fix the third tower. The green data is much more widely spread in each dimension making it hard to distinguish from the others in terms of parameter settings, but we can note that it does have lower values of t_2 that pull its associated regression curve downward in the third coordinate plot from the top in the center image of Figure 42. The magenta crystal represents a single point which is its own local maximum. This local maximum point has a low t_0 and t_1 value, a moderate t_2, but a very late t_3.

5.4 Conclusions

In this section, we investigate the use of a topology-based clustering technique on the data generated by dynamic event tree methodologies. Our clustering technique is inspired by Morse-Smale complex which partitions the data based on their uniform gradient behavior. We demonstrate our analysis technique and visual interface on a dataset generated for a sodium-cooled fast reactor during an aircraft crash scenario, and describe some interesting insights obtained from its end state and transient analysis. Combining with previously obtained mean-shift clustering results, and with further involvement by domain scientists, we expect to obtain in-depth understandings of such datasets.

In comparison to mean-shift clustering, each cluster has a representative center, but in an optimization or goal-seeking setting, a maximum/minimum representative of each cluster could be more beneficial to domain experts which is something that is readily available and inherent to topological clustering. It is also important to note that new topology-based clustering techniques could be inspired by the various topological structures such as contour trees and Reeb graphs. However understanding whether and how such techniques could produce meaningful results for analyzing DPRA-related nuclear datasets remains a challenge.

6 A BWR SBO Case Study

6.1 Overview

In this section, we describe some preliminary results for a nuclear simulation dataset using our analysis and visualization tool. The dataset is part of the Risk Informed Safety Margin Characterization (RISMC) Boiling Water Reactor (BWR) station blackout (SBO) case study. We investigate the use of a topology-based clustering technique on such a dataset. The clustering technique we utilizes focuses on a domain-partitioning algorithm based on topological structures known as the Morse-Smale complex, which partitions the data points into clusters based on their uniform gradient flow behavior. We believe such analysis and visualization present the user with views that could be considered complementary to more standard clustering techniques of the data. Such an analysis could help illuminate key features that may be otherwise hidden using traditional techniques. We assume some basic understanding of our tool and methodology, see Section 2 for technical details.

6.2 Background on BWR SBO Scenario

The analysis considered a **BWR Mk. I** system during a Loss of Offsite Power (LOOP) event followed by loss of the diesel generators (DGs), i.e. station blackout (SBO). In more details:

- At time $t = 0$: the following events occur:
    - LOOP condition occurs: due to external event, the off-site 500 KV line de-energize and the switchyard connected to the 161 KV line gets damaged
    - LOOP alarm triggers the following actions:
        - Operators successfully scram the reactor put it in sub-critical conditions by fully inserting the control rods in the core
        - MSIVs successfully close and isolate the primary containment from the turbine building
        - Emergency DGs successfully start and the 4160 V busses are kept energized
        - Decay heat generated by the core is removed from the pressure vessel through the residual heat removal system
        - DC systems (i.e., batteries) are functional
- SBO condition occurs: due to internal failure, the set of DGs fails and the 4160 busses de-energize. Removal of decay heat is impeded. Reactor operators start the SBO emergency procedures and perform:
    - RPV level control using RCIC or HPCI
    - RPV pressure control using SRVs
    - Containment monitoring (both drywell and PSP)

- Plant staff start recovery operations to bring back on-line the DGs while the recovery of grid power is underway by the grid owner company emergency staff
- Due to loss of AC power, the Component Cooling Water (CCW) and Service Water (SW) systems cannot provide cooling for the seal of the circulation pumps. The overheating of these seals can cause their failure causing release of water from the RPV to the drywell (pump seal LOCA)
- Due to the limited life of the battery system and depending on the use of DC power, battery power can deplete. When this happens, all remaining control systems are offline causing the reactor core to heat until maximum temperature limit for the clad is reached: core damage (CD) condition occurs
- If DC power is still available and one of these condition are reached:
  - Failure of both RCIC and HPCI
  - HCTL limits reached
  - Low RPV water level

  then the reactor operators activate the ADS system in order to depressurize the RPV and allow firewater injection, if available
- When AC power is recovered, through successful re-start/repair of DGs or off-site power, RHR can be now employed to keep the reactor core cool.

6.3 Analysis Methodology

Overview. We apply clustering algorithms based on the Morse-Smale complex to a nuclear dataset obtained from a BWR System SBO scenarios. We model such a dataset as a high-dimensional scalar function f defined on a finite set of points $\mathbb{X}$ in $\mathbb{R}^n$. We partition the points in $\mathbb{X}$ based on their function values and gradient behavior with respect to the approximated Morse-Smale complex. That is, points belong to the same cluster if they have uniform gradient flow behavior. We further the analysis by obtaining a topological summary for each cluster and try to infer the correlations between simulation parameters and system observations. For a detailed description of our tool and clustering methodology, see Section 3 and Section 5.

In this study, the domain scientists are interested in what combination of conditions (in the form of input simulation parameters) can cause potential reactor failure (i.e. nuclear meltdown witnessed by maximum core temperature exceeding a threshold value). Furthermore, they are interested in the impact of increased reactor power on safety of the nuclear plant, in terms of time required for various recovery procedures.

Raw data description. An ensemble of 19996 transient simulations has been generated, and among which 6597 scenarios are considered system failures when the reactor breaches a failure temperature threshold before the end of simulation (i.e. power is recovered or the firewater is aligned). The rest of the 13399 scenarios are considered simulation completions/success. Each simulation includes information regarding the timing of various recovery attempts (e.g. cooling recovery, fire water, etc.), the reactor power level and the failure temperature threshold. There are 12 input parameters[7], which encode uncertainty by sampling from a continuous range during simulation:

- **FailureTimeDG**: Failure time of Diesel generators (DGs). This corresponds to the time of SBO event.
- **RecoveryTimeDG**: Recovery time of DGs.
- **OFFsitePowerRecoveryTime**: Offsite AC power recovery time.
- **SRV1stuckOpenTime**: The time when 1 Safety Relief Valve (SRV) is stuck open.
- **SRV2stuckOpenTime**: The time when 2 or more SRVs are stuck open.
- **cladFailureTemperature**: Uncertainty in failure temperature for the clad is characterized by a triangular distribution having: a lower limit of 1800 F (PRA success criterion); a upper limit of 2600 F (Urbanic-Heidrick transition temperature); and a mode of 2200 F (10CFR regulatory limit).
- **HPCIFailToRunTime**: The time when High Pressure Core Injection (HPCI) fails to run.
- **RCICFailToRunTime**: The time when Reactor Core Isolation Cooling (RCIC) fails to run.
- **ReactorPower**: Reactor power uprate. It is sampled between $1 = 100\%$ and $1.2 = 120\%$, e.g. a 1.2 reactor power means that the reactor is operating under 120% power (e.g. with 120% of raw material).
- **ADSactivationTimeDelay**: Manual Automatic Depressurization System (ADS) activation. Operator manually depressurizes the Reactor Pressure Vessel (RPV) by activating the ADS system. This parameter actually measures the time delay from the HCTL event, not the time from 0 to when ADS is activated.
- **firewaterTime**: Firewater injection (as an emergency action, when RPV pressure is below 150 psi, plant staff can connect the firewater system to the RPV in order to cool the core and maintain an adequate water level).
- **TotalBatteryLife**: (BatteryLife plus ExtendedECCSoperationBattery) Battery life combined with Extended ECCS operation (that is, operators may extend RCIC/HPCI and SRVs control even after the batteries have been depleted. They manually control RCIC/HPCI by acting on the steam inlet valve of the turbine and/or supply DC power to the SRVs through spare batteries).

All the above time-related parameters are measured from the time of the SBO event (in seconds), which is the FailureTimeDG. The output variable obtained from the transient simulation is **maxCladTemp**, which is the maximum clad temperature reached during the entire course of the simulation. Such a variable is less than or equal to the failure temperature threshold (e.g cladFailureTemperature). In addition, an additional discrete variable **outcome** is given to classify success ($\text{outcome} = -1$) from failure ($\text{outcome} = 1$) scenarios.

[7] These input parameters are chosen to be analyzed as they are the only uncertain parameters under consideration in our current context.

Data standardization. The above data is pre-processed with a standardization process. We employ the Z-score scaling as discussion in Section 3, which is a data standardization process so that all dimensions are on the same scale. Recall for values of each dimension, we subtract the mean and divide by the standard deviation. Although more robust approaches are available [41], the above technique is often sufficient.

Reduce number of dimensions. Upon further observations of the nature of the simulation, we further transform the data by reducing the number of dimensions. In particular, we introduce 3 new dimensions by eliminating 6 dimensions from the raw dataset:

- **ACPowerRecoveryTime**: $\min\{\text{RecoveryTimeDG}, \text{OFFsitePowerRecoveryTime}\}$.
- **SRVstuckopen**:] $\min\{\text{SRV1stuckopen}, \text{SRV2stuckopen}\}$.
- **CoolingFailtoRunTime**: $\max\{\text{HPCIFailToRunTime}, \text{RCICFailToRunTime}\}$.

Data representation (9D and 8D). With data pre-processing, we arrive at several representations of the data. Currently we consider absolute timing (w.r.t. the time of the SBO event) for all timing-related measurements (and later we may consider relative timing w.r.t. sequential events). We model the dataset as either 9*D* or 8*D* scalar functions.

The 9*D* cases include the following input variables: (1) FailureTimeDG; (2) ACPowerRecoveryTime; (3) SRVstuckOpenTime; (4) cladFailureTemperature; (5) CoolingFailtoRunTime; (6) ReactorPower; (7) ADSactivationTimeDelay; (8) firewaterTime; and (9) TotalBatteryLife. The output variable is the maxCladTemp (MT). We further analyze these functions based on all scenarios, system failure scenarios and simulation success scenarios. A particular 8*D* case arises for the failure scenarios, where we consider only 8 of the 9 dimensions (excluding the parameter cladFailureTemperature as it equals to the output parameter of choice, maxCladTemp).

Alternatively, we would like to understand how the ReactorPower is correlated with other input parameters. We consider ReactorPower (RP) as the output variable and replace the input parameter ReactorPower with maxCladTemp, creating new 9D functions. Again, when considering failure scenarios, we analyze the 8D function excluding cladFailureTemperature as the input (since it equals to the maxCladTemp).

Data representation (7D). Since the above dimensions are all related to timing measurements except cladFailureTemperature and ReactorPower, our 7*D* case drops these two dimensions. We have run new simulations to generate new 7*D* datasets with 4997 scenarios (i.e. points) by fixing the values for these two parameters. The analysis here then focus on understanding more accurately the correlations among timing parameters and the output. The output variables considered are: (1) maxCladTemp (MT); (2) the end simulation time (EST), e.g. how quickly the system failures occur.

Experimental cases. We list all of our experiments as follows, for a total of 8 cases:

9D : 9D-MT-all, 9D-MT-success; 9D-RP-all, 9D-RP-success.

8D : 8D-MT-failure; 8D-RP-failure.

7D : 7D-MT-all; 7D-EST-failure.

6.4 Preliminary Results Under Z-score Scaling

For each of the above 8 cases, we showcase the preliminary results with (a) topological summary; (b) inverse-coordinate plots; and (c) distribution of input parameters. We would like to know: (a) the general, approximated topological structure of the data; (b) how input dimensions help differentiate the clusters; (c) which dimensions are more importantly correlated to the output; and (d) the profiles (distributions) of input parameters associated with each cluster.

For subsequent sections, the shorthand description of each case encodes the dimension, output variable of interest, scenarios included and number of crystals obtained. For example, 9D-MT-all-3C means a 9D function under analysis, where: the 9 dimensions are what have been discussed before; the output variable is MT; the analysis covers all scenarios (both successes and failures); and the obtained result includes 3 crystals (which correspond to 3 clusters of the dataset). Some input/output parameters have been abbreviated in the analysis/visualization interface, e.g. FWTime stands for firewaterTime, etc.

Initial Observations. Before we start our detailed topology-based analysis, we obtain some initial observation of the data across multiple dimensions based upon some initial visualizations, as shown in Figure 43, where we begin to see trends linking individual input dimensions with the maxCladTemp or ReactorPower over the full 19996 scenarios. For example, in Figure 43 (top right), we look at the pairwise scatter plots across all dimensions (e.g. for a total of $9 \times 8/2 = 36$ scatter plots) where the points are colored by maxCladTemp. There appears to be a direct correlation between the maxCladTemp and the ACPowerRecoveryTime (second column or first row). This means that if we recover AC power quickly, then the maximum temperature the simulation reaches will likely be lower. Similarly, if DCPowerLife is high, then the resulting maxCladTemp is more likely to be lower (see the bottom row).

On the other hand, the inverse coordinate plots (Figure 43 (top left), where for each plot, the X-axis corresponds to MT and the Y-axis is the individual input dimension) indicate that neither of these two input parameters are necessary or sufficient to achieve a low maxCladTemp, since the data points span the entire range of MT for all values of DCPowerLife and ACPowerRecoveryTime.

Similar observations using ReactorPower could be made. However, our goal is to understand not individual parameters in isolation, but correlations between a high-dimensional input parameter space and the output. By looking at the topological summary of the dataset and clustering the points based on their approximated gradient behavior, we hope to highlight specific areas of the domain that are topologically significant. Such areas correspond to particular combinations of input parameter settings that may be interesting for further investigation by domain scientists. We hope such topology-based techniques could offer complementary views for the structure of the dataset, which may lead to new insights.

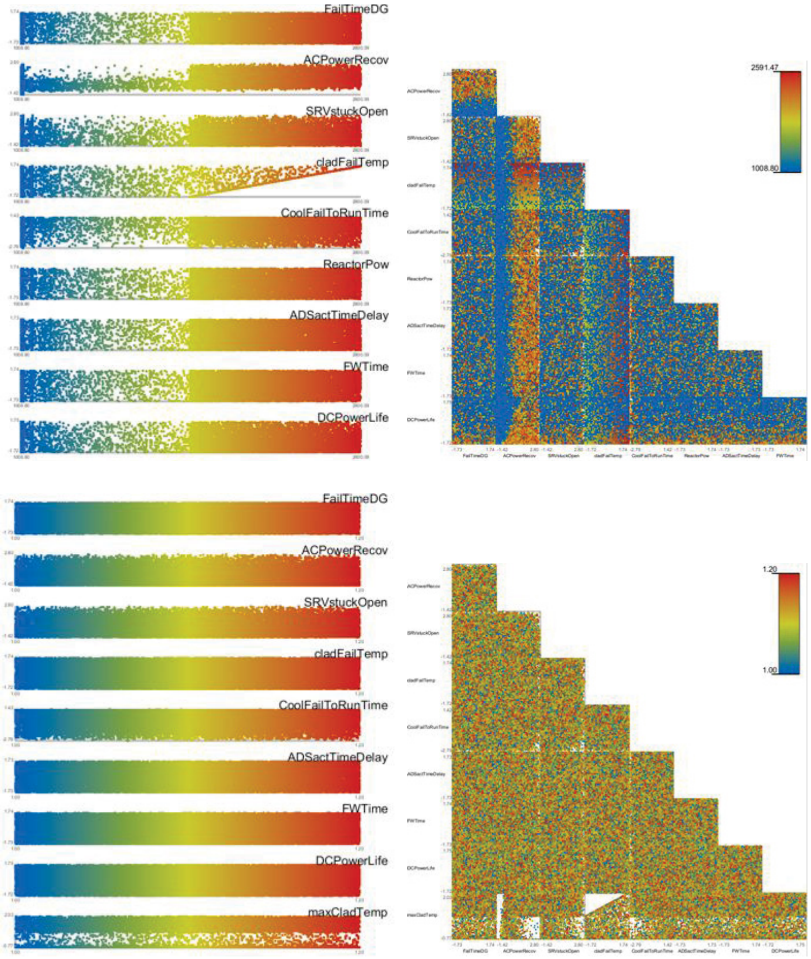


Figure 43: Initial observation of the data points. Top row: inverse coordinate plots (left) and pairwise scatterplots of data with maximum clad temperature (MT) mapped to the color scale. Bottom row: inverse coordinate plots (left) and pairwise scatterplots of data with reactor power (RP) mapped to the color scale.

6.4.1 9D-MT-all-3C

Preliminary results are shown in Figure 44 and Figure 45. As in most subsequent cases, there is one large crystal (a.k.a. cluster) that encompasses the majority of the data, and a few smaller crystals that share either a global minima or maxima with this large crystal. Since the largest crystal (red) covers the ranges of each input dimension, we focus our analysis on the smaller crystals and study how they differ from the largest crystal.

As shown in Figure 44 (left), the three crystals share one global minimum maxCladTemp value of 1008.80 (in Kelvin) and diverge to 3 different local maxima. The red crystal attains the highest value of 2600.09. The green crystal reaches a maximum value of 2597.20. The blue crystal reaches a maximum of 2534.16, which is not significantly lower than the other two.

Now we focus on the inverse coordinate plots in Figure 44 (right) with projected regression curves that summarize the data values within each crystal (without points projection). In particular, the regression curves for FailureTimeDG and the ADSactivationTimeDelay could be used to differentiate the three crystals at their maxima. The regression curves for ACpowerRecoveryTime, cladFailTemp, and FWtime vary slightly across crystals, indicating some consistent correlations between these parameters and the output parameter, maxCladTemp. The DCPowerLife, ReactorPower, and CoolFailToRunTime are similar for the two smaller crystals (blue and green) and differ from the larger crystal (red), indicating a possible interesting subspace of lower dimensional setting necessary to achieve high maxCladTemp.

It is important to note for all our analysis involving cladFailureTemperature (e.g. cladFailTemp in the visualization), that it may introduce unwarranted topological artifact by capping the maximum temperature the simulation reaches (therefore leading to plateau within the landscape of the high-dimensional function without reaching a true local maxima).

The distribution of input parameters across all crystals (clusters) are shown as histograms in Figure 45. Here, we separate the time-related attributes (left column) with others (e.g. remaining columns: cladFailureTemperature; maxCladTemperature; ReactorPower). We further separate failure scenarios with success scenarios using darker and lighter hues respectively. These histograms further help us distinguish differences among the three crystals. For example, FailTimeDG (light gray/dark gray bars in the left column) has higher range of values for the green crystal, compared to the blue crystal; while the blue crystal favors higher values of SRVstuckOpen (light yellow/dark yellow) when compared to the other two crystals.

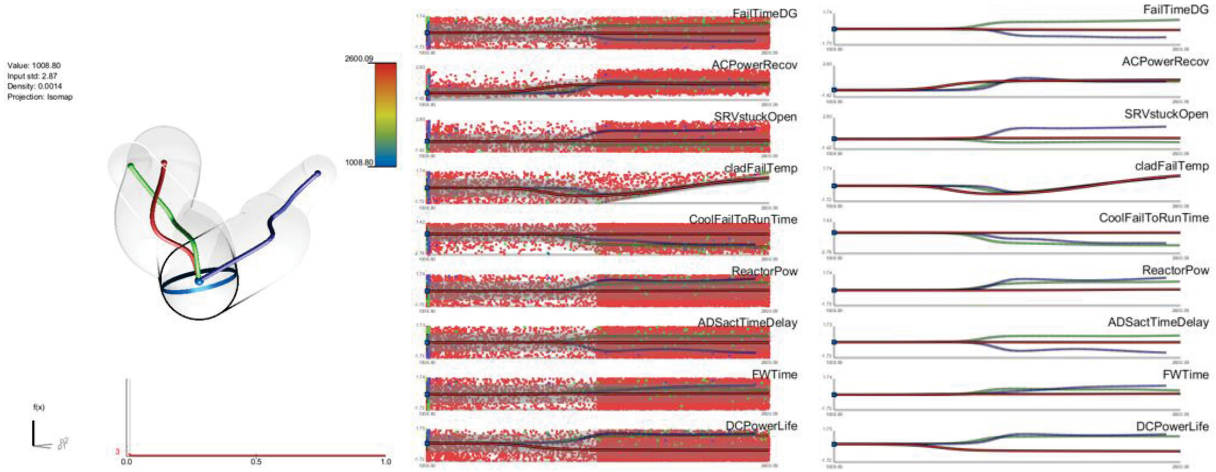


Figure 44: **9D-MT-all-3C**: topological summary (left), inverse coordinate plots with (middle) and without (right) points projection. Points and regression curves are colored by cluster (crystal) memberships.

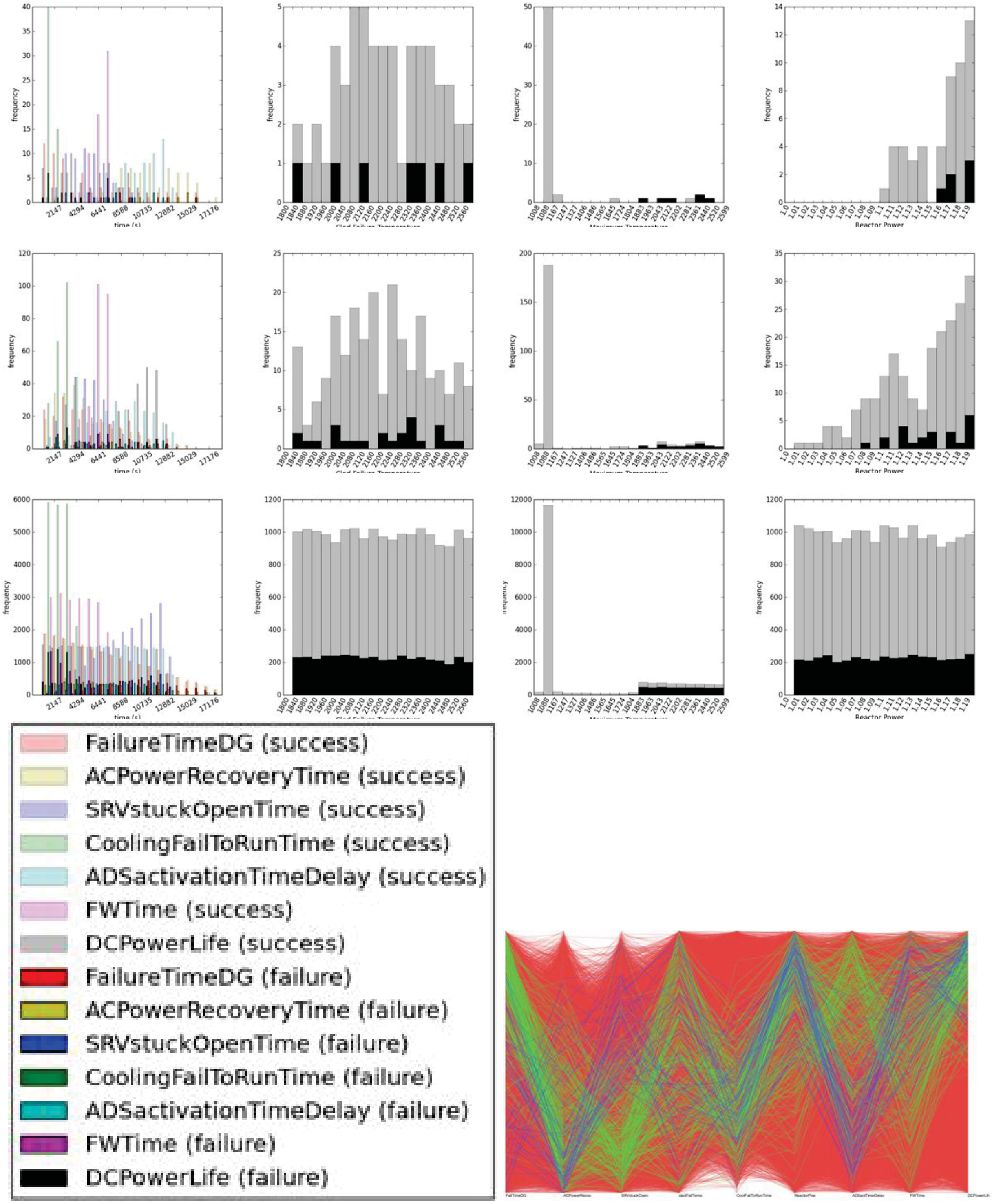


Figure 45: **9D-MT-all-3C**: distribution of input parameters across different clusters. First three rows: blue, green and red clusters respectively. From left to right: distribution of 7 time-related input parameters; cladFailureTemperature; maxCladTemperature; ReactorPower. Last row: histogram legends for all 7 time-related input parameters for both failures and successes scenarios (left); parallel coordinate plots for all three clusters (right).

6.4.2 9D-MT-success-4C

Preliminary results are shown in Figure 46 and Figure 47. In Figure 46 (middle), we see the dense, red crystal covers majority of the data points whose corresponding output values do not have a clear trend with respect to any single input parameter. In other words, the data is spread over all dimensions at all temperature levels in this crystal, and so the regression curve is more or less constant with respect to each dimension. In the topological summary view (Figure 46 (left)), the global minima connects all four crystals and is represented by a maxCladTemp of 1008.80, whereas the local maxima differ more than in the previous example. Their values, in decreasing order are: 2591.47 (red), 2509.11 (magenta), 2444.34 (green), and 2051.76 (blue).

In Figure 46 (right), the regression curves for cladFailTemp and ACPowerRecoveryTime are consistent in shape across all crystals. Regression curves for CoolingFailToRunTime and FWTime best differentiate the four crystals at their corresponding maxima. FailureTimeDG shows similar behavior for the three smaller crystals while the large crystal (red) has a significantly lower value for high maxCladTemp output. Regression curves for ADSactivationTimeDelay, DCPowerLife, SRVstuckOpen time, and ReactorPower differentiate at most two smaller crystals from the red crystal. Such topological summary of the data indicates that each local maximum varies with respect to multiple parameters and no single parameter exists to differentiate them. On the other hand, it implies that there maybe lower-dimensional subspaces formed by some of the input parameters.

Again, the histograms of Figure 47 further help differentiate different crystals. For instance, for the blue crystal, CoolFailToRunTime (blue bar) has a mode of higher value than the other three, and that the ReactorPower has a distribution that is more dense on the lower values than the other three crystals.

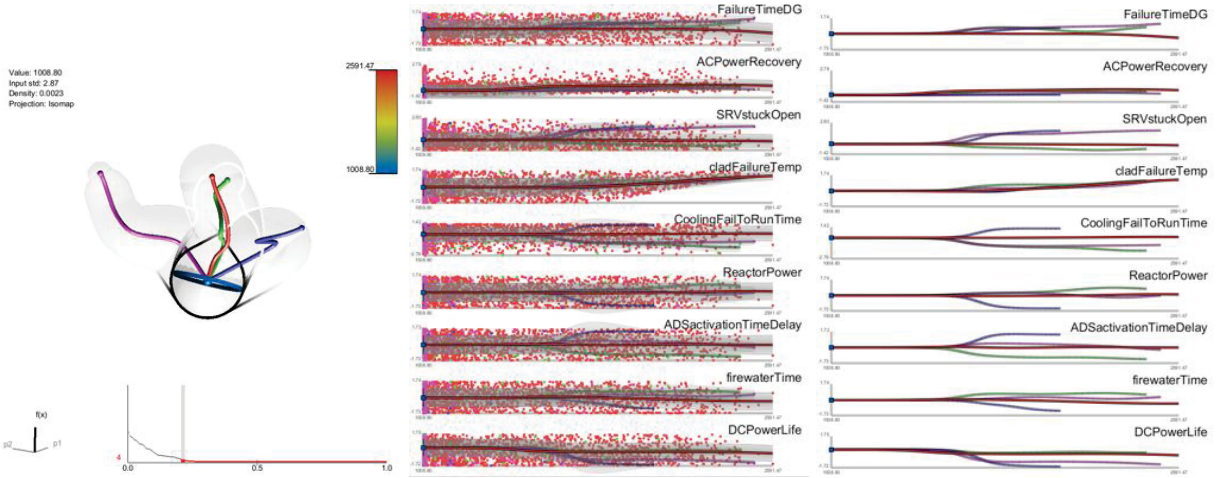


Figure 46: **9D-MT-success-4C**: topological summary (left), inverse coordinate plots with (middle) and without (right) points projection.

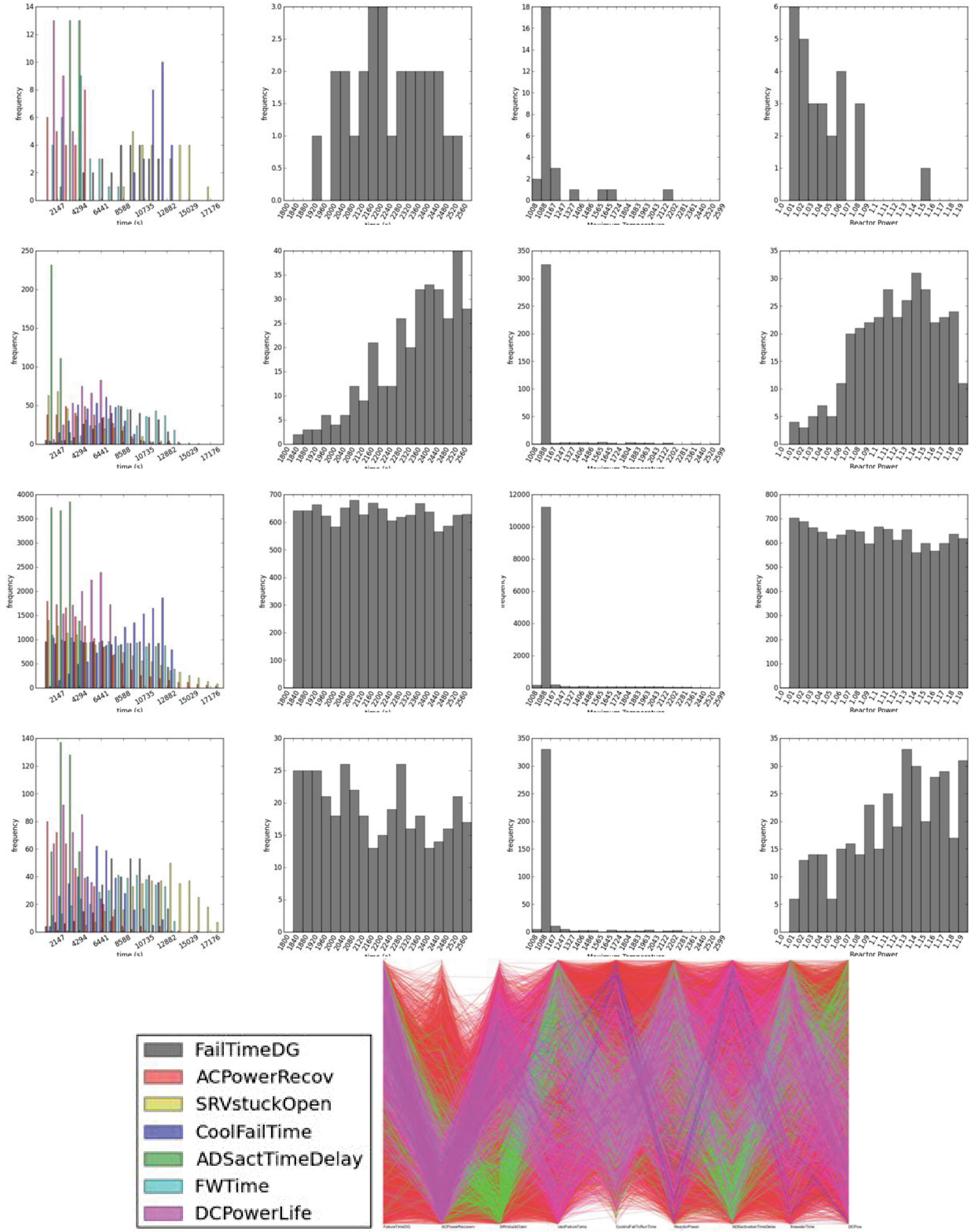


Figure 47: **9D-MT-success-4C**: distribution of input parameters across different clusters. First four rows: blue, green, red, and magenta clusters respectively. From left to right: distribution of 7 time-related input parameters; cladFailureTemperature; maxCladTemperature; ReactorPower. Last row: histogram legend for all 7 time-related input parameters (left); parallel coordinate plots for all four clusters (right).

6.4.3 8D-MT-failure-3C

Figure 48(left) shows a topological summary that includes a red crystal (with majority of the data points). However, it also includes a separate local minimum in addition to a separate local maximum. The shared local minimum between the blue and red crystals has a value of 1800.11, whereas the minimum for the green crystal is valued at 1800.37. The shared local maximum between the red and green crystals attains a value of 2600.06, whereas the blue crystal has a slightly lower local maximum value at 2593.00.

In Figure 48 (right), the ACPowerRecovery does not vary much across crystals, as it reaches comparable values at both ends of the regression curves. Such consistency may indicate weak correlation between ACPowerRecovery with maxCladTemp (despite our initial observations of the entire dataset). On the other hand, FailTimeDG, ReactorPower, and ADSactivationTimeDelay could independently differentiate all three crystals, whereas DCpowerLife, SRVstuckOpen, and CoolFailToRunTime could differentiate one crystal from the others.

Figure 49 offers more detailed differentiation among the three crystals by looking at the distribution of input parameters. For instance, looking at the ReactorPower (right column) plots, we see two distinct distribution patterns for the blue and green crystals and a uniform-like distribution for the red crystal. Furthermore, in the leftmost column, we can see the blue crystal is represented by a lower CoolFailToRunTime (blue bar) than the green crystal. The red crystal has values at both, but has more points near the green crystal. This matches our observation from Figure 48.

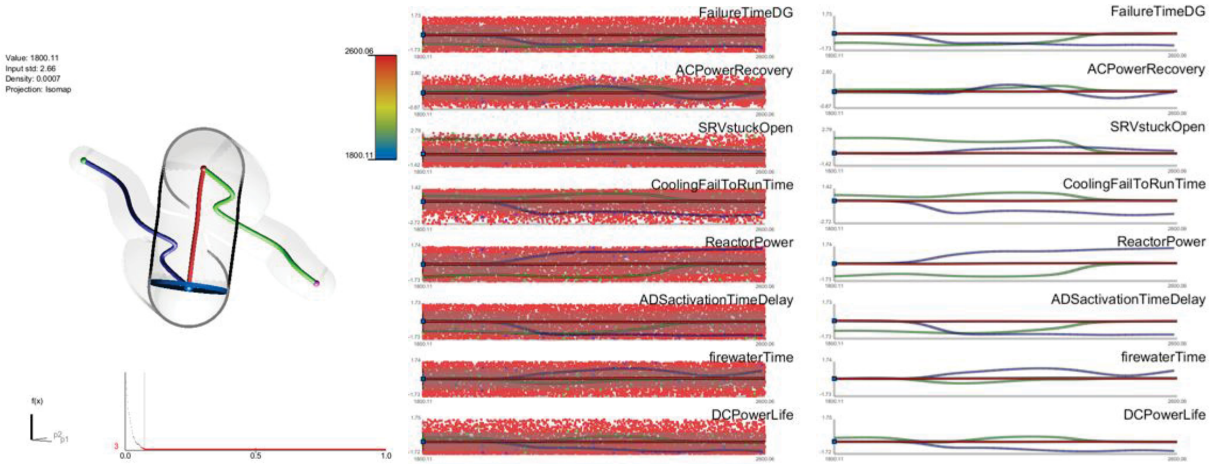


Figure 48: **8D-MT-failure-3C**: topological summary (left), inverse coordinate plots with (middle) and without (right) points projection.

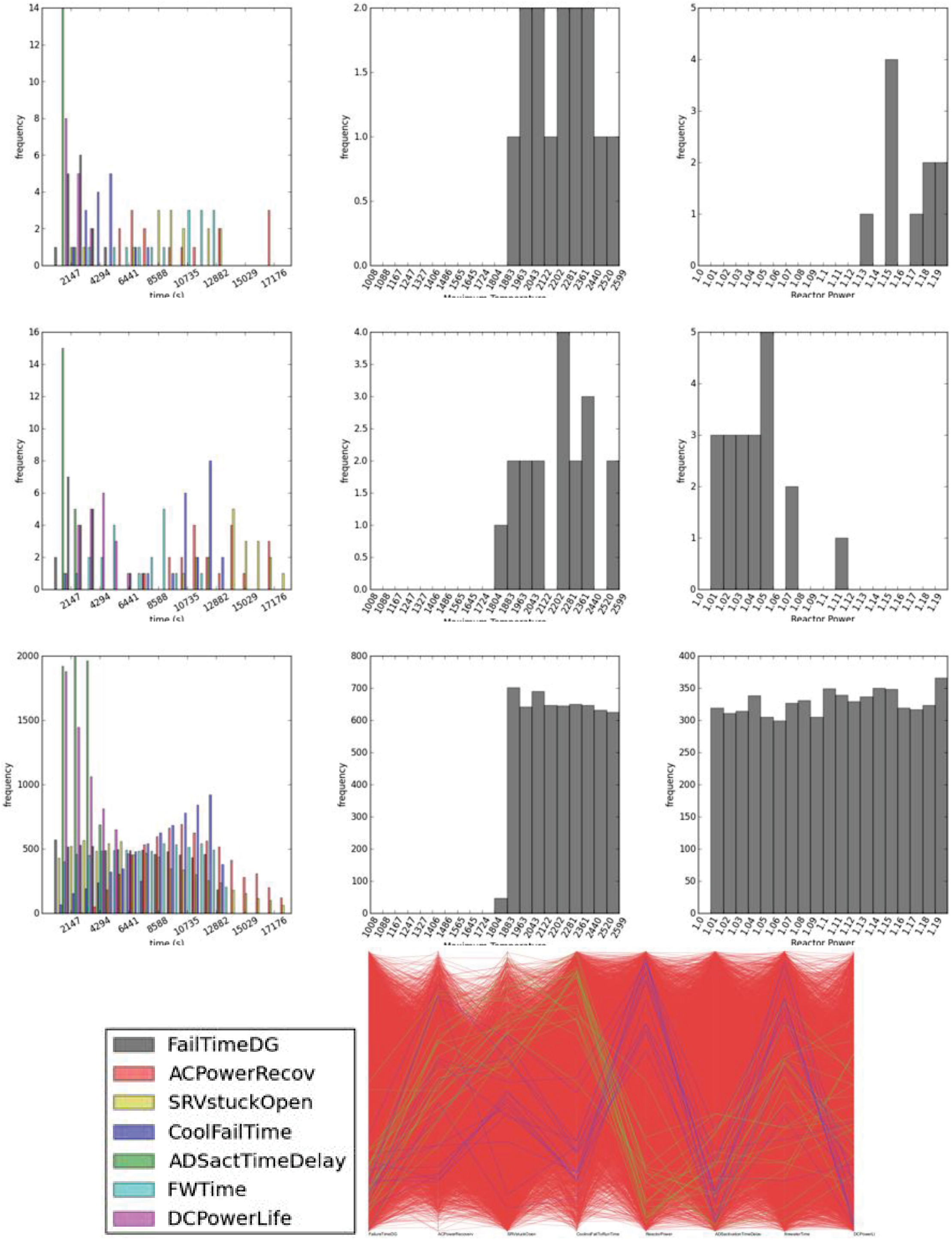


Figure 49: **8D-MT-failure-3C**: distribution of input parameters across different clusters. First four rows: blue, green, and red clusters respectively. From left to right: distribution of 7 time-related input parameters; maxCladTemperature; ReactorPower. Last row: histogram legend for all 7 time-related input parameters (left); parallel coordinate plots for all three clusters (right).

6.4.4 9D-RP-all-4C

For the next three subsections, we swap the input parameter of ReactorPower with the output parameter of maxCladTemp, since the domain scientists are interested in optimizing the amount of ReactorPower upscaling without a significant increase to the maxCladTemp. Thus, in such a setting, multiple local maxima would indicate areas of interest for maximizing the ReactorPower, especially in analyzing just the success scenarios. We begin by considering all scenarios in this section.

In Figure 50 (left), we have selected a topological summary at a particular scale where all four crystals share a global minimum value of 1 for ReactorPower. The individual local maxima values are 1.20 for the red, blue, and green crystals, whereas the magenta crystal maxes out at 1.19. The magenta crystal has very few points (only 2), but actually represents a persistent/significant feature.

By analyzing the inverse coordinate plots in Figure 50 (right), we see that the parameters that could distinguish the four crystals at or near their maxima are CoolFailToRunTime and FWTime. The maxCladTemp, DCPowerLife, cladFailTemp, FailureTimeDG, ACPowerRecoveryTime, and ADSactivationTimeDelay differentiate at most three crystals from each other (in different combinations).

The histograms of Figure 51 highlight two crystals (green and magenta in Figure 50 (left)) that contain no failure scenarios. This is a new insight that may help us understand which input parameters are correlated with the ReactorPower. The green, magenta, and blue crystals all share high CladFailTemp values, compared to the red crystal. In addition, the former three crystals contain no intermediate values for maxCladTemperature, indicating the core is either functioning under low temperature; or has failed.

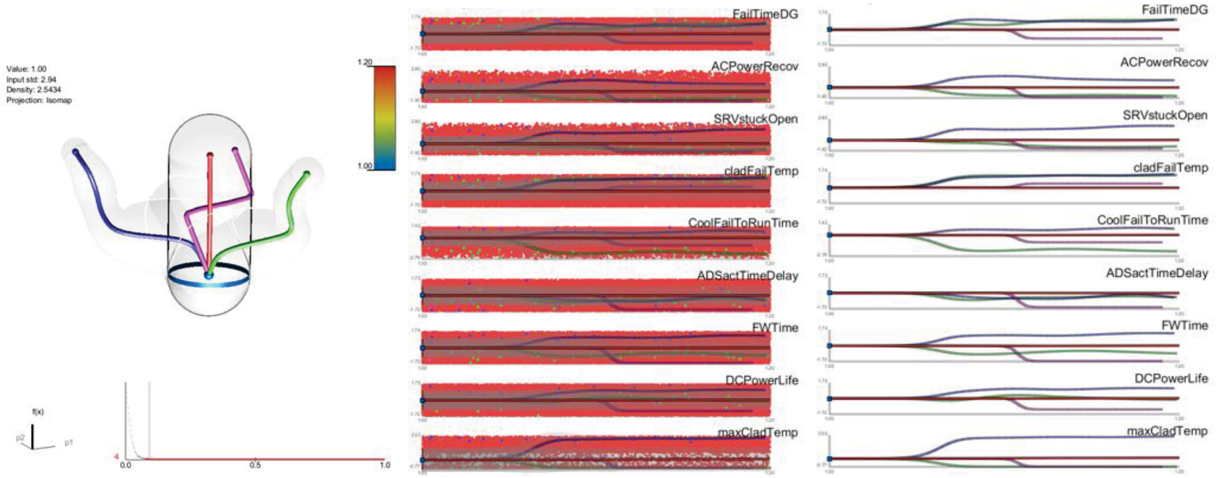


Figure 50: **9D-RP-all-4C**: topological summary (left), inverse coordinate plots with (middle) and without (right) points projection.

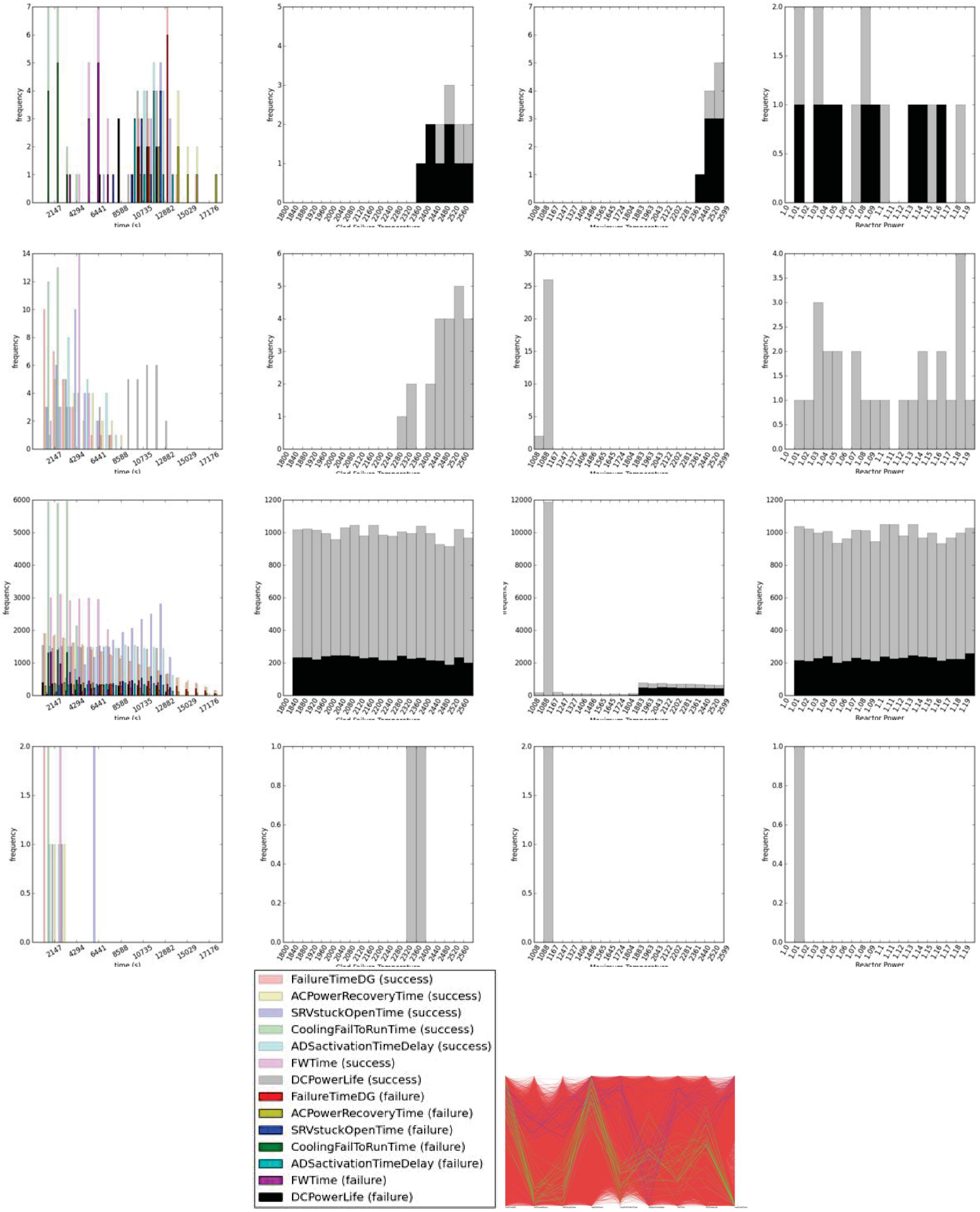


Figure 51: **9D-RP-all-4C**: distribution of input parameters across different clusters. First four rows: blue, green, red, and magenta clusters respectively. From left to right: distribution of 7 time-related input parameters; cladFailureTemperature; maxCladTemperature; ReactorPower. Last row: histogram legend for all 7 time-related input parameters (left); parallel coordinate plots for all four clusters (right).

6.4.5 9D-RP-success-4C

At a chosen scale, we obtain four crystals (clusters) where the topological summary contains three distinct local maxima and two local minima, as shown in Figure 52 (left). The shared maxima has a ReactorPower valued at 1.20, while the shared minima has a value of 1.00. The green crystal descends to a local minimum value of 1.01, whereas the blue and magenta crystals ascend to maxima values of 1.20 and 1.19, respectively.

In the inverse coordinate plot of Figure 52 (right), the regression curves for maxCladTemp is constant across all values of ReactorPower, as such an input parameter has no influence on the ReactorPower since we are only looking at success scenarios. Other inverse coordinate plots have at least one regression curve overlapping with another, implying that a combination of parameters could be used to differentiate the crystals. The regression curves for the green crystal across all dimensions, on the other hand, deviates most from other crystals.

By shifting our attention to histograms in Figure 53, we see some defining patterns for the blue (top row) and green (second row) crystals. The blue crystal has a relatively low CoolFailToRunTime and FWTime, whereas the green crystal has higher values for these attributes. Conversely, the blue crystal has a high cladFailTemp, while the green crystal has a a lower cladFailTemp.

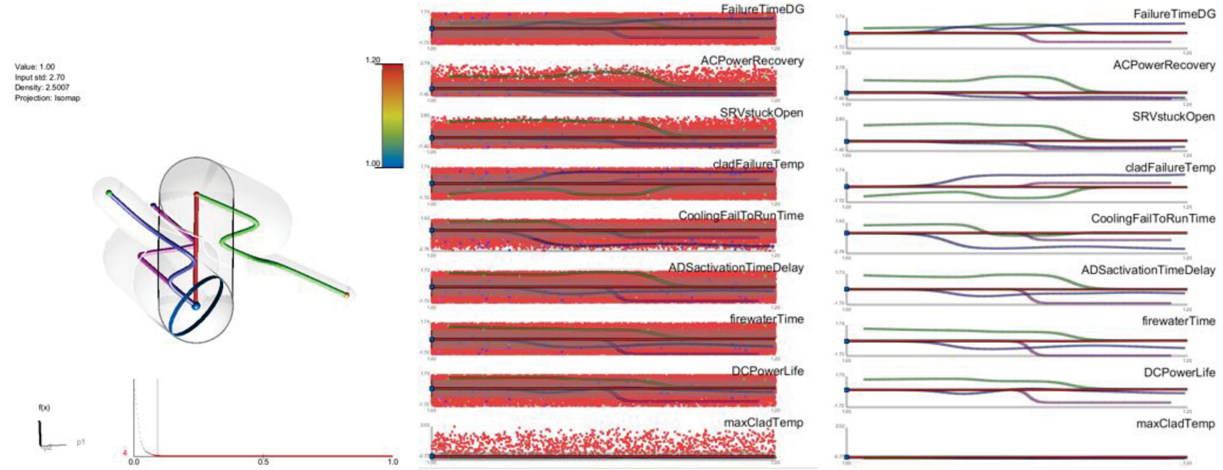


Figure 52: **9D-RP-success-4C**: topological summary (left), inverse coordinate plots with (middle) and without (right) points projection.

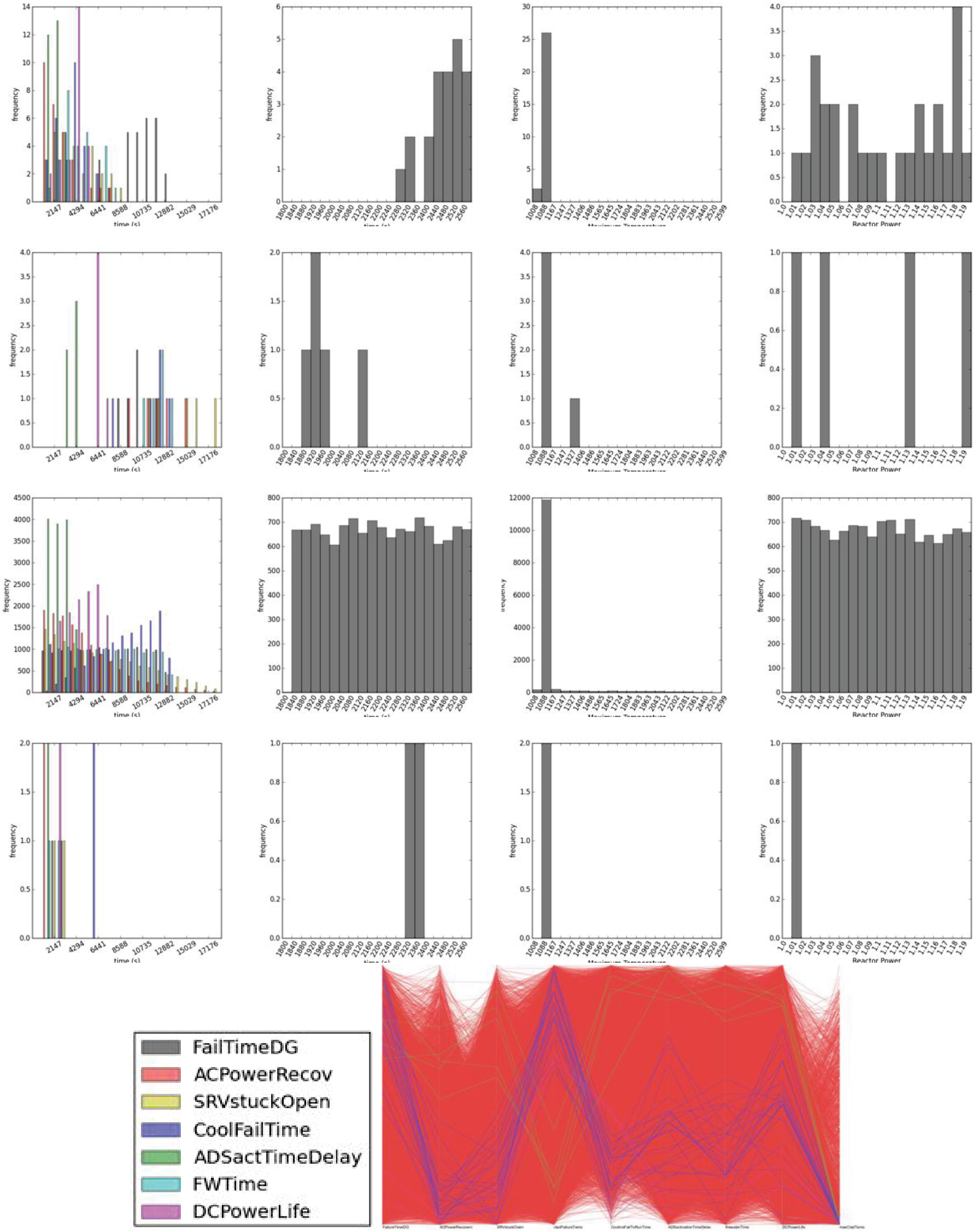


Figure 53: **9D-RP-success-4C**: distribution of input parameters across different clusters. First four rows: blue, green, red, and magenta clusters respectively. From left to right: distribution of 7 time-related input parameters; cladFailureTemperature; maxCladTemperature; ReactorPower. Last row: histogram legend for all 7 time-related input parameters (left); parallel coordinate plots for all four clusters (right).

6.4.6 8D-RP-failure-4C

For analysis that focuses only on failure scenarios, we choose a topological summary with four crystals that contains 3 local maxima and 2 local minima, as shown in Figure 54 (left). The shared minima has a value of 1.00. The green crystal attains the same minimum value, but in a distinct location in the domain. All maxima have values of 1.20, except for the magenta crystal with a lower maximum of 1.18.

In the inverse coordinate plots of Figure 54 (right), FWTime, DCPowerLife, and maxCladTemp can independently be used to differentiate the four crystals. ADSactivationTimeDelay, ACPowerRecoveryTime, CoolFailToRunTime, and FailureTimeDG separate three of the four crystals, and so a combination of these parameters could be used to distinguish the four crystals. SRVstuckOpenTime separates the blue and magenta crystals from the red and green, creating two branches of the regression curves near the maxima.

In Figure 55, the distributions of ACPowerRecoveryTime have different modes for the blue, green and magenta crystals, which may be useful in distinguishing these areas of the domain.

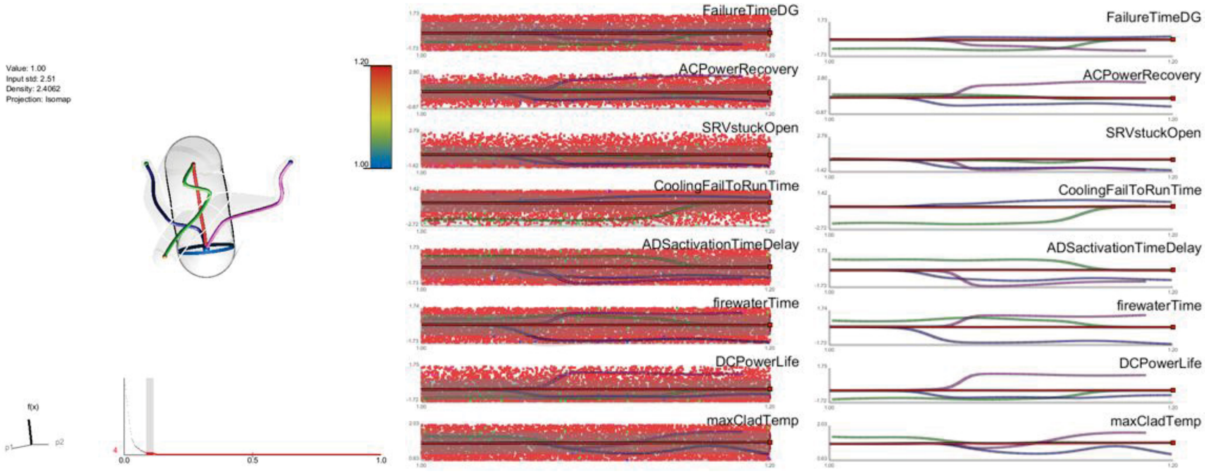


Figure 54: **8D-RP-failure-4C**: topological summary (left), inverse coordinate plots with (middle) and without (right) points projection.

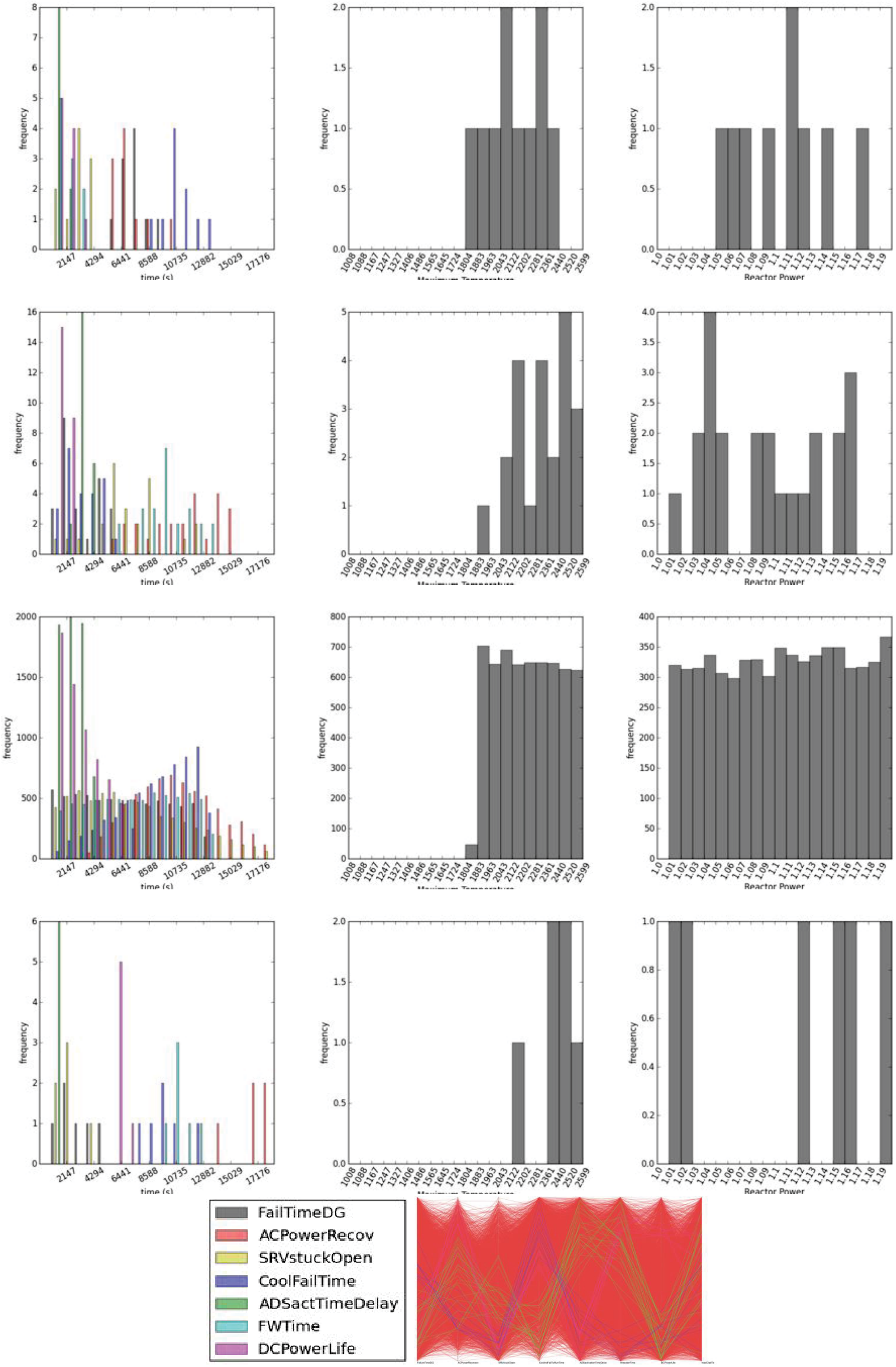


Figure 55: **8D-RP-failure-4C**: distribution of input parameters across different clusters. First four rows: blue, green, red, and magenta clusters respectively. From left to right: distribution of 7 time-related input parameters; maxCladTemperature; ReactorPower. Last row: histogram legends for all 7 time-related input parameters (left); parallel coordinate plots for all four clusters (right).

6.4.7 7D-MT-All-3C

Preliminary results are shown in Figure 56 and Figure 57. In this example, we see a topological structure characterized by a global maximum temperature of 2200.12 connecting two local minimum temperature values of 1008.80 and 1008.85 occurring in distinct parameter settings of the domain. There is also a smaller local maximum represented by the blue crystal that attains a value of 1401.79, but first we will discuss how these two local minima are separated in the domain. The FailureTimeDG, CoolingFailToRunTime, and ADSactivationTimeDelay all show similar settings at or near the two local minima. The minimum of the green crystal is separated from the other shared minimum by a noticeably later FWTime. In addition, a shorter DCPowerLife and SRVstuckOpenTime and a slightly later ACPowerRecoveryTime are characteristic of this minimum compared to the minimum of the red and blue crystals' shared minimum.

Lastly, the third blue crystal represents a cluster composed entirely of success cases which we can see from the left image in Figure 57. The distinctive parameters of note are the higher values of DCPowerLife, ADSActivationTimeDelay, and FailureTimeDG leading to the distinct maximum point in domain space.

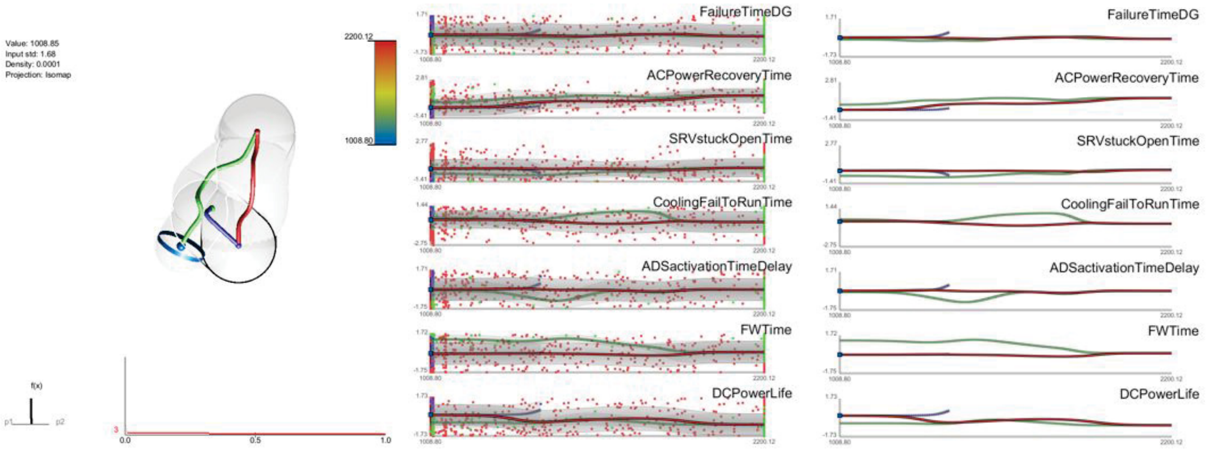


Figure 56: **7D-MT-All-3C**: topological summary (left), inverse coordinate plots with (middle) and without (right) points projection. Points and regression curves are colored by cluster (crystal) memberships.

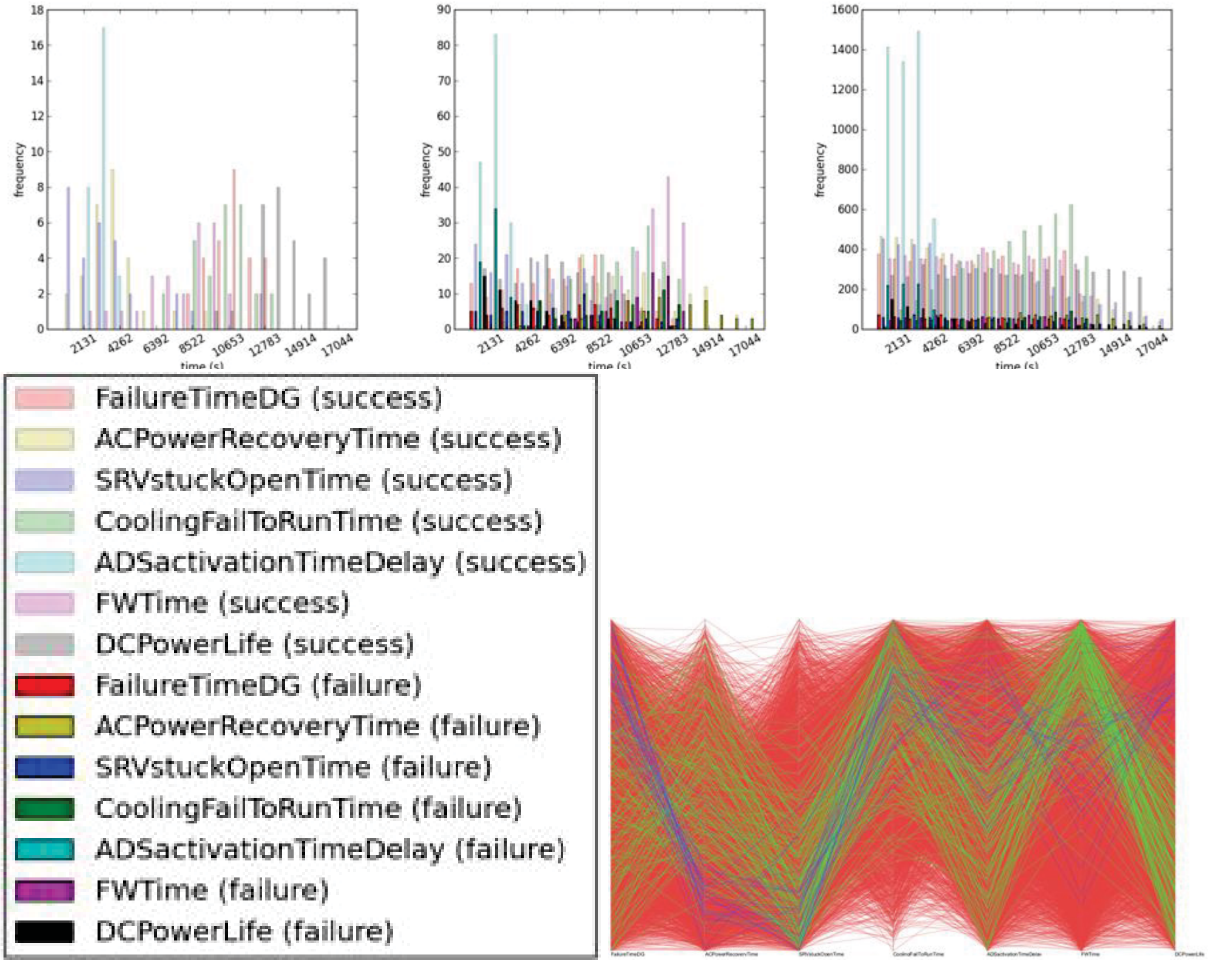


Figure 57: **7D-MT-all-3C**: distribution of input parameters across different clusters. Top row: blue, green and red clusters respectively. Bottom row: histogram legends for all 7 time-related input parameters for both failures and successes scenarios (left); parallel coordinate plots for all three clusters (right).

6.4.8 7D-EST-Failure-4C

Preliminary results are shown in Figure 58 and Figure 59. In this dataset, we consider only failure cases and use as the output value the end simulation time (EST), that is how quickly the system failure occurred. From the left image in Figure 58, we note that there is a global fast failure case marked by a time of 434.82 seconds. All failure cases can be traced down to this single, global minimum case, but from it arise four distinct local maxima. One interpretation is to look at the local maxima as possible near successes since they represent the latest time for failure. In other words, the temperature growth of these scenarios was slower and thus allowed more time to be contained.

The four maxima have significantly different EST values, in increasing order they are: 14229.26 seconds (magenta), 16581.58 (blue), 18985.33 (red), and 22084.97 (green). From the center and right plots of Figure 58, we note that parameters like CoolingFailToRunTime and DCPowerLife offer little distinction amongst the regression curves of different color. The blue crystal in particular allows us the least ability to differentiate it, in many cases overlapping with either the green or red curve in most dimensions. However, FWTime and FailureTimeDG both show good separation between the separate crystals at or near their respective maxima.

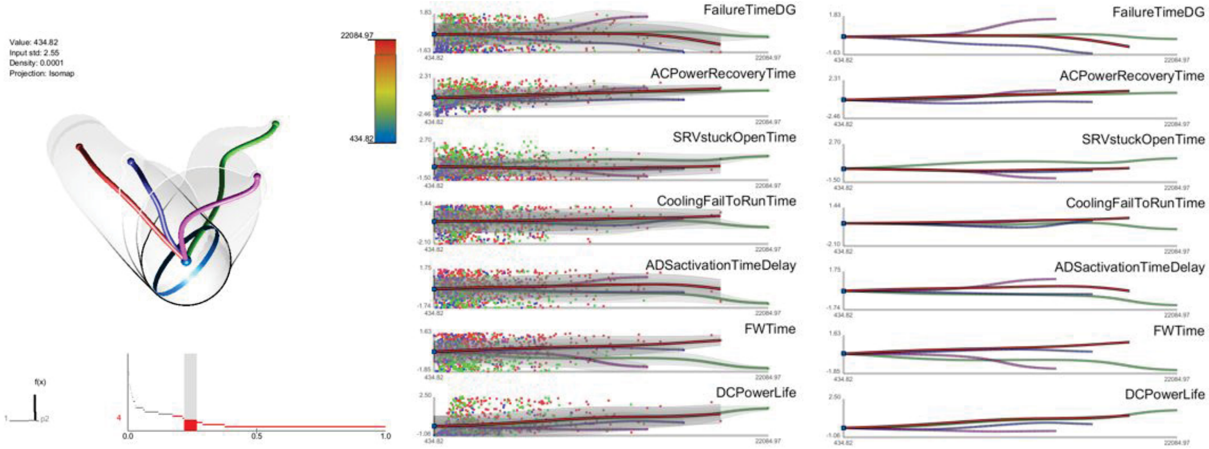


Figure 58: **7D-EST-Failure-4C**: topological summary (left), inverse coordinate plots with (middle) and without (right) points projection. Points and regression curves are colored by cluster (crystal) memberships.

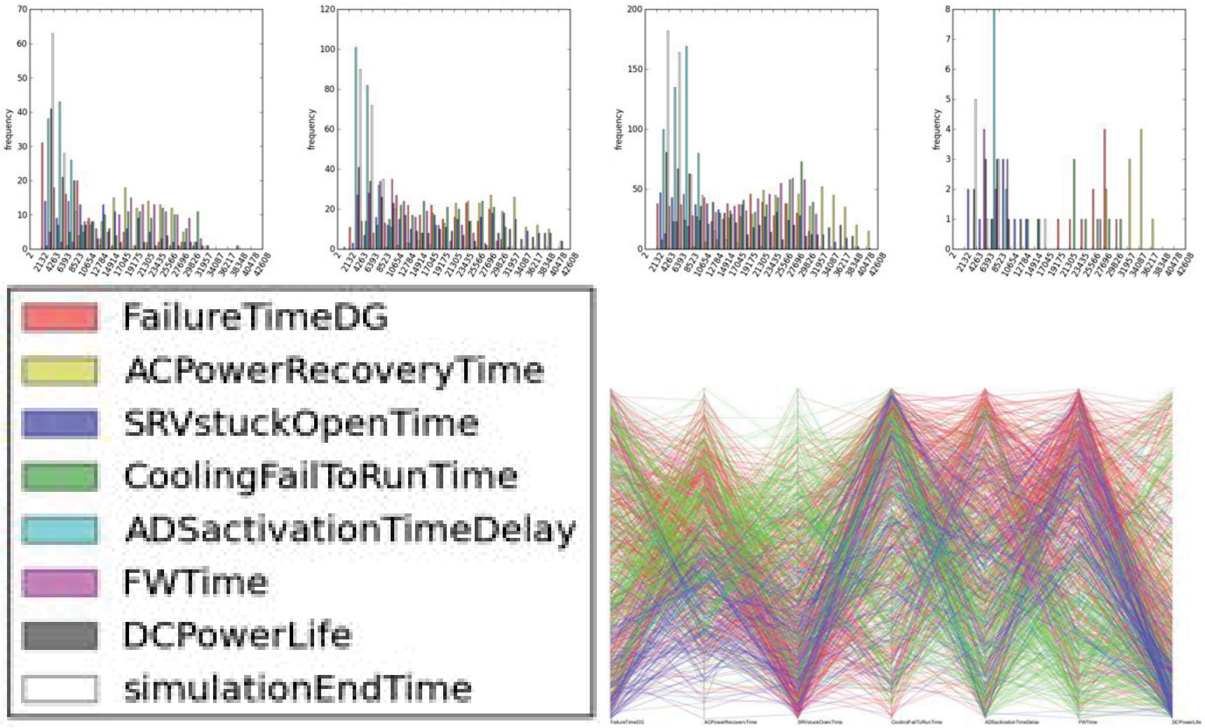


Figure 59: **7D-EST-Failure-4C**: distribution of input parameters across different clusters. Top row: blue, green, red and magenta clusters respectively. Bottom row: histogram legends for all 7 time-related input parameters for both failures and successes scenarios (left); parallel coordinate plots for all three clusters (right).

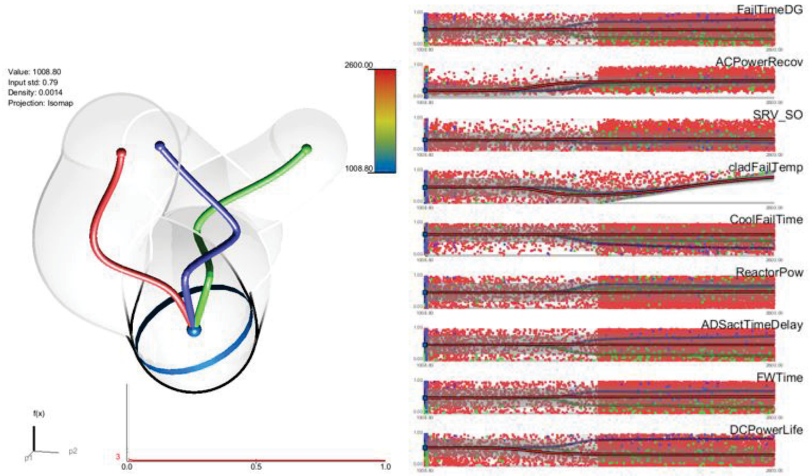


Figure 60: **9D-MT-All-3C**: Left: Topological skeleton projected into a 3D view, right: inverse coordinate plots with summary regression curves superimposed per dimension and crystal.

6.5 Preliminary Results Under $[0, 1]$-Scaling

For completeness, we also present results under $[0, 1]$-scaling of the data instead of the Z-score scaling.

6.5.1 9D-MT-All-3C

As shown in Figures 60 and Figure 61, we see that the majority of the data points belong in a single crystal. We view the 3-crystal setting where they share a common global minimum valued at 1008.80. The three maximum values are quite close in value: 2595.24 (green), 2597.11 (blue), and 2600.00 (red). In the right image of Figure 60, we see that four dimensions show the same behavior for all three crystals: ACPowerRecoveryTime, SRVStuckOpenTime, CladFailureTemp, and ReactorPower. DCPowerLife splits the blue crystal into higher values separating it from the green and red crystals. The remaining parameters of ADSActivationTimeDelay, FailureTimeDG, FWTime, and CoolingFailToRunTime each show distinct settings for the 3 crystals at or near their respective maximum values. From the histograms in Figure 61, we see that the first two rows (the blue and green crystals), have distinct values of CoolingFailToRunTime. The green crystal has a high frequency of early ADSActivationTimeDelay values that are not present in the other crystals.

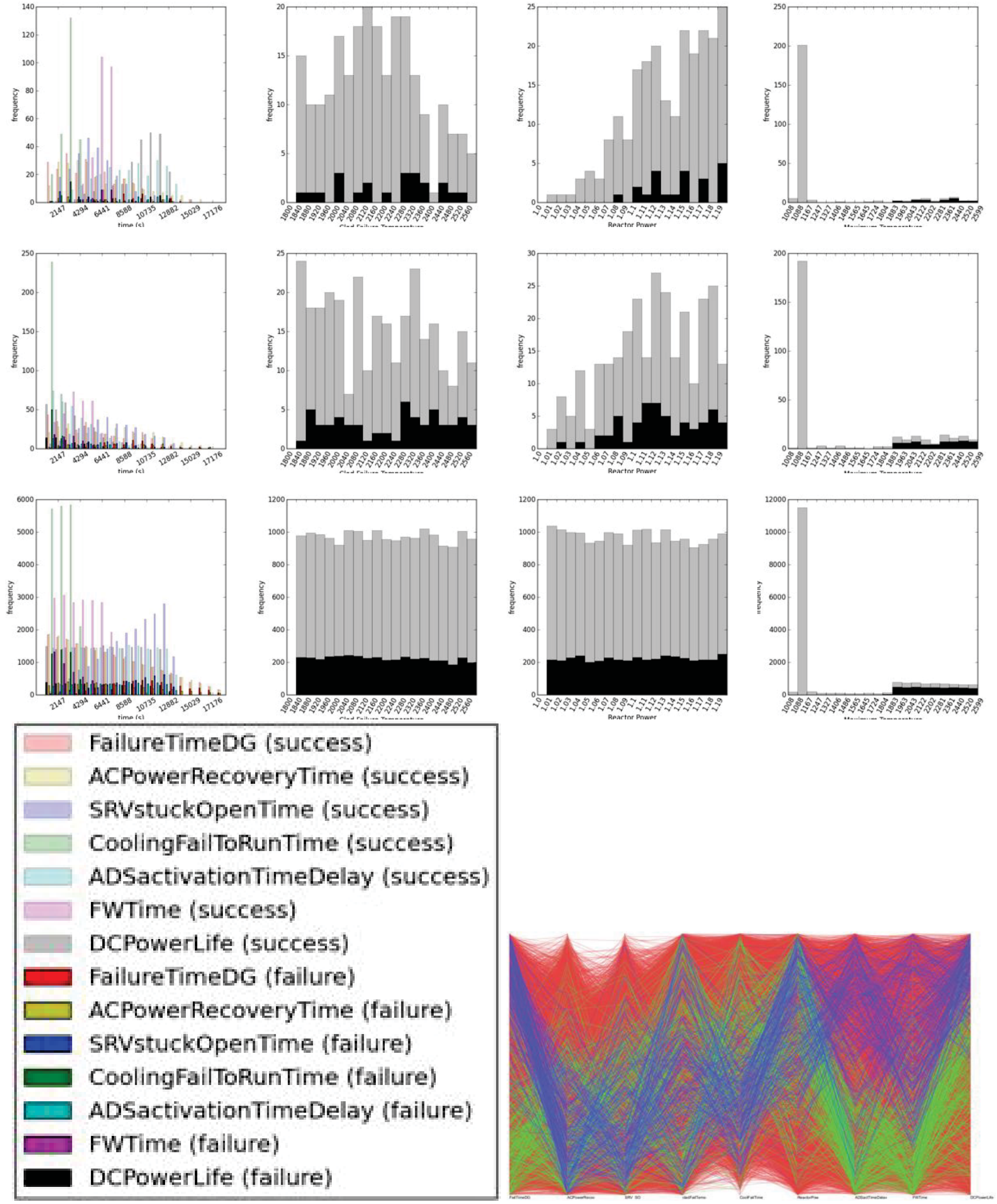


Figure 61: **9D-MT-All-3C**: distribution of input parameters across different clusters. Top row: blue crystal, Second row: green crystal, Third row: red crystal. The first column includes all time-based parameters, whereas the second column has cladFailutreTemperature data, the third column contains ReactorPower data, and the final column is the output: MaximumCladTemperature. Bottom row: histogram legends for all 7 time-related input parameters for both failures and successes scenarios (left); parallel coordinate plots for all three clusters (right).

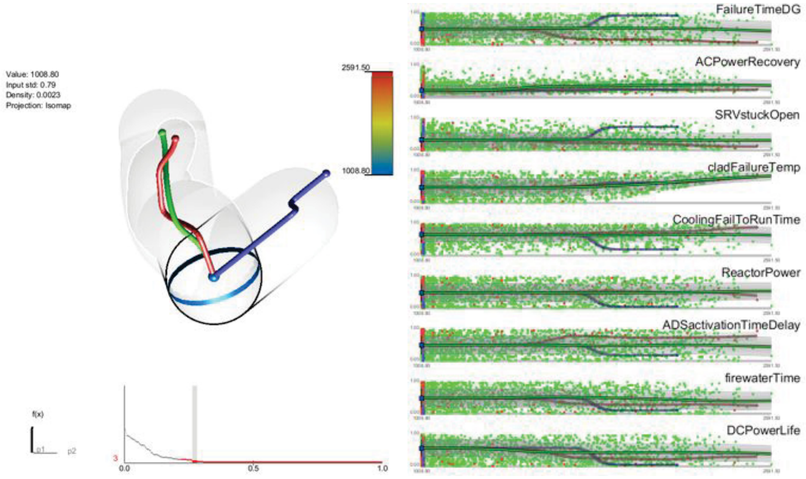


Figure 62: **9D-MT-Success-3C**: Left: Topological skeleton shown in a 3D view, right: inverse coordinate plots with regression curves of each cluster superimposed per dimension.

6.5.2 9D-MT-Success-3C

We consider only success scenarios. Figures 62 and 63 highlight a case with three crystals. These crystals share a global minimum valued at 1008.80 and diverge to three distinct maxima with values of 2165.97 (blue), 2526.02 (red), and 2591.50 (green). Of the 13399 success scenarios, 13004 belong to the green crystal, the red crystal has another 384, and the blue crystal is made up of only 11 scenarios. As in prior cases, if we look at the inverse coordinate plots, we can attempt to distinguish the features that separate the crystals from one another. In this case, cladFailureTemp and ACPowerRecovery show regression curves that follow similar paths. However, FailureTimeDG and ADSActivationTimeDelay can be used to separate all three crystals. In addition, SRVStuckOpenTime, CoolingFailToRunTime, DCPowerLife, ReactorPower, and FWTime, could be considered as differentiating factors for the blue crystal from the red and green crystals. By looking at the histograms in Figure 63, one can see the blue crystal has a high concentration of scenarios with an early SRVStuckOpenTime compared to the red and green crystals.

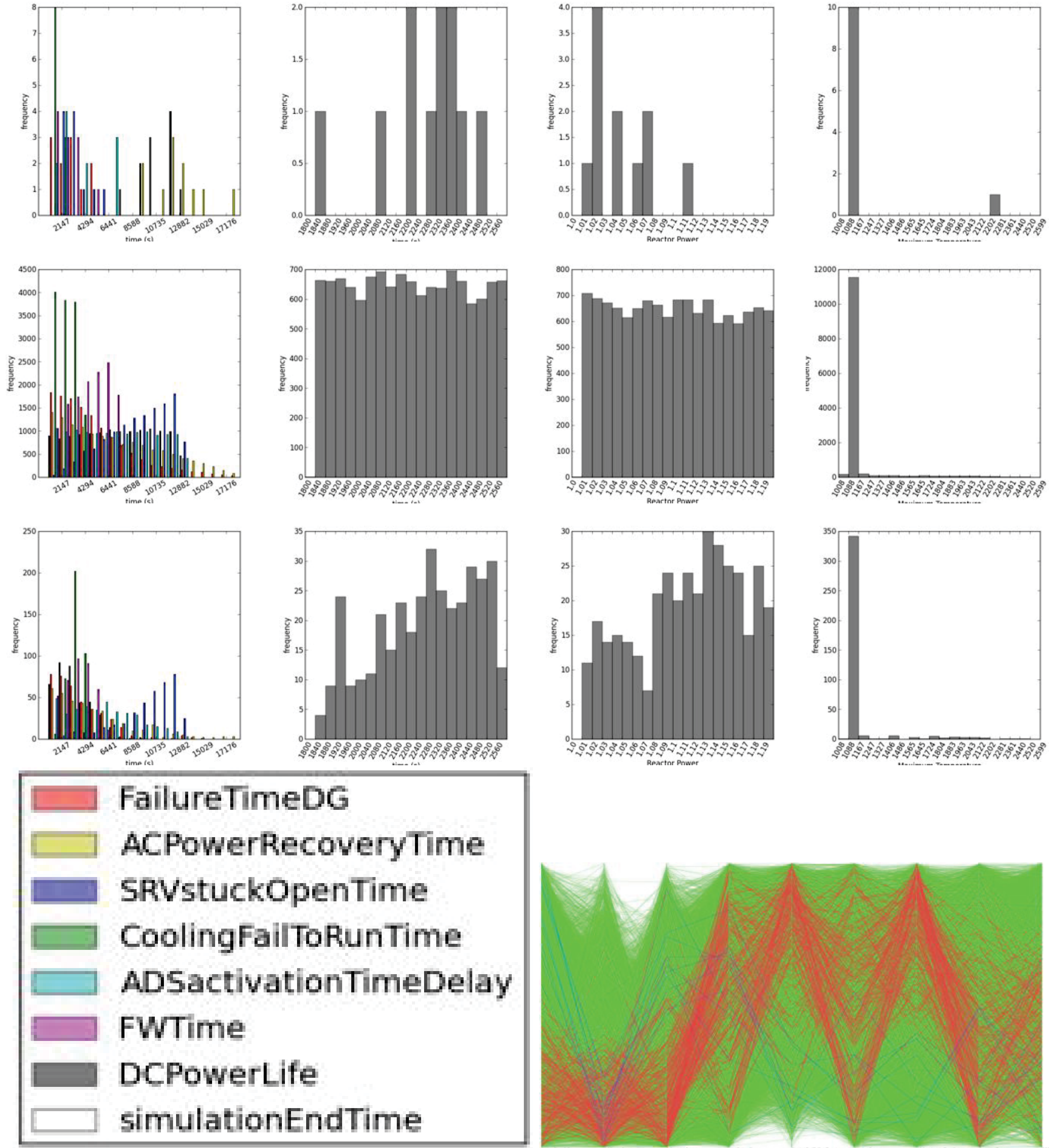


Figure 63: **9D-MT-Success-3C**: distribution of input parameters across different clusters. Top row: blue crystal, second row: green crystal, and third row: red crystal. The first column includes all time-based parameters, whereas the second column has cladFailutreTemperature data, the third column contains ReactorPower data, and the final column is the output: MaximumCladTemperature. Bottom row: histogram legends for all 7 time-related input parameters for success scenarios (left); parallel coordinate plots for all three clusters (right).

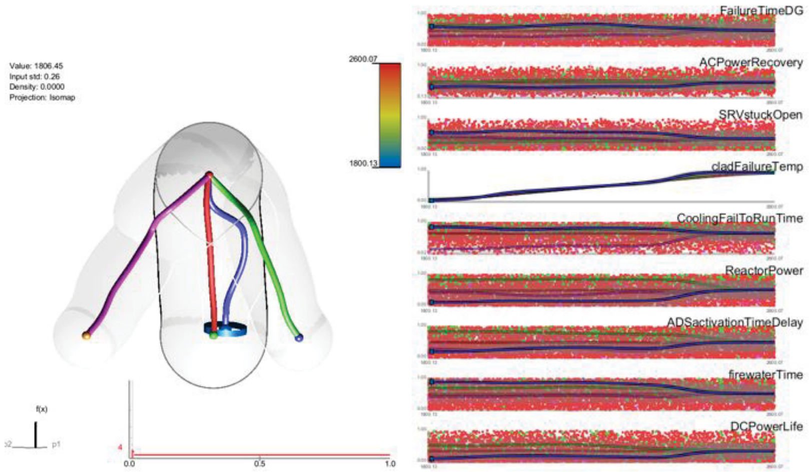


Figure 64: **9D-MT-Failure-4C**: Left: Topological skeleton shown in a 3D view, right: inverse coordinate plots with regression curves of each crystal superimposed per dimension.

6.5.3 9D-MT-Failure-4C

Switching our focus to only the failure scenarios, we arrive at a setting with a single, global maximum valued at 2600.07 and four separate minima. All four minima have similar values: 1800.13 (red), 1800.43 (green), 1802.28 (magenta), and 1806.45 (blue). Most of the data resides in the red crystal which has data spanning the entire range of each input dimension, thus its regression curve is more or less a straight line down the center of each input coordinate plot in the right image of Figure 64. From the inverse coordinate plots, it is hard to identify a single dimension that distinguishes all four crystals, although 5 of the dimensions differentiate at least 3 different modes occurring at the left ends of the plots. ReactorPower distinguishes the green crystal with a high value, the red and magenta crystals with moderate values, and the blue crystal with a lower value. Similar characteristics occur for ADSActivationTimeDelay, that is, it has a high value in the green crystal, moderate value in the red crystal and low value in the blue crystal. DCPowerLife exhibits a similar pattern only with all four crystals shifted downward slightly. FWTime has the blue crystal with a high value at the minimum end (left), the green crystal sits below the blue, but above the red and magenta crystals which are nearly the same. Lastly, the CoolingFailToRunTime has the blue and green crystals both exhibiting high values, the red crystal slightly below them, and the magenta crystal distinguishing itself with a low value of CoolingFailToRunTime. In the histograms of Figure 65, we note that the magenta crystal has a high concentration of early SRVstuckOpenTime and ADSActivationTimeDelay values. The blue crystal has ReactorPower values concentrated on the lower end, whereas the green crystal has concentration on the high end.

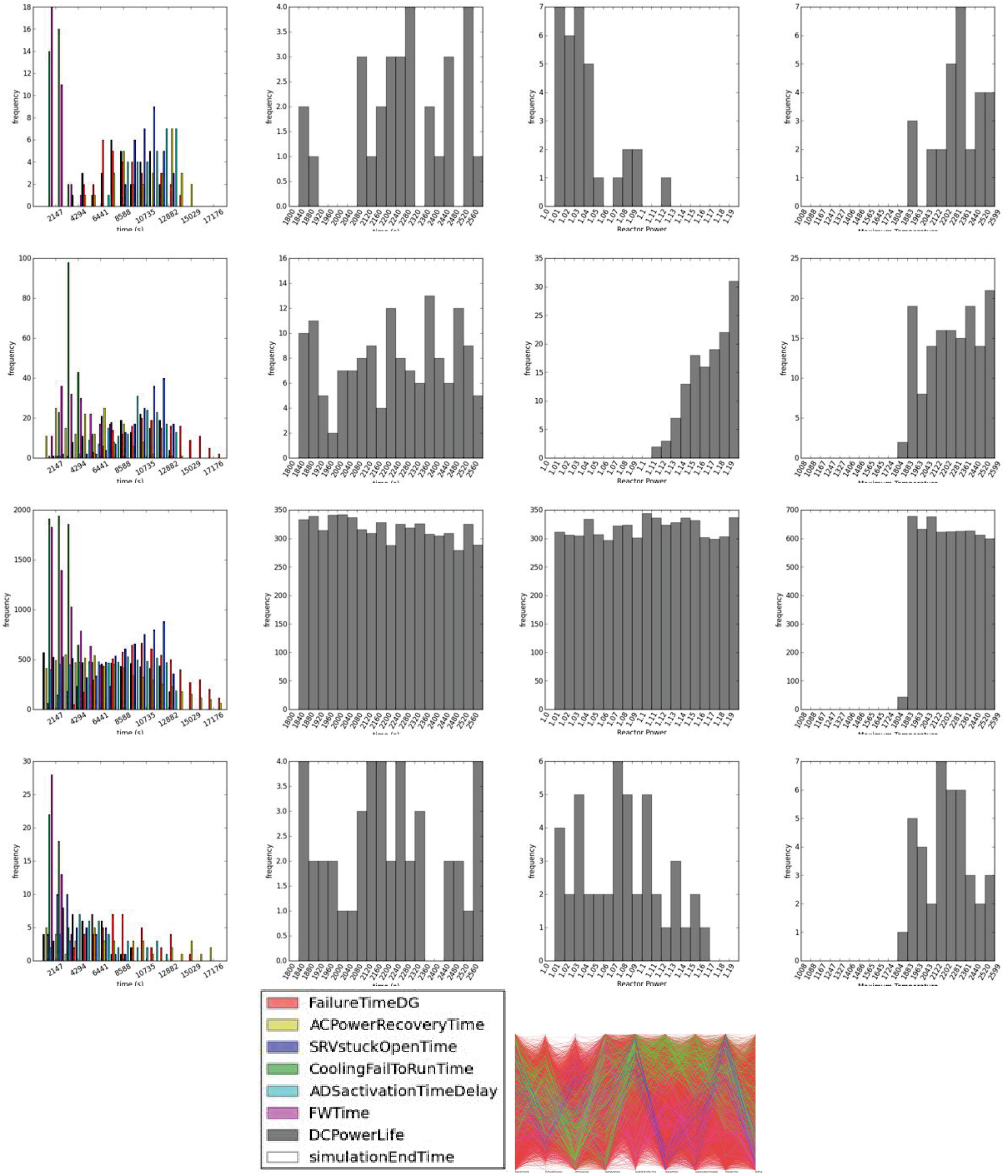


Figure 65: **9D-MT-Failure-4C**: distribution of input parameters across different clusters. Top row: blue crystal, second row: green crystal, third row: red crystal, and fourth row: magenta crystal. The first column includes all time-based parameters, whereas the second column has cladFailutreTemperature data, the third column contains ReactorPower data, and the final column is the output: MaximumCladTemperature. Bottom row: histogram legends for all 7 time-related input parameters for failures scenarios (left); parallel coordinate plots for all four clusters (right).

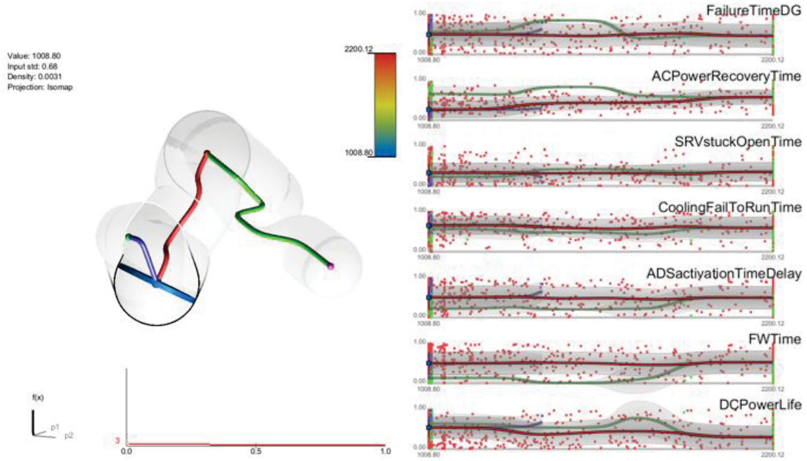


Figure 66: **7D-MT-All-3C**: Left: Topological skeleton. Right: inverse coordinate plots showing points colored by cluster id and their regression curves drawn with a gray transparent tube representing the standard deviation with respect to each input dimension.

6.5.4 7D-MT-All-3C

We visualize three crystals in the topological summary, characterized by one crystal with large number of data points shown in red (Figure 66), which shares its maximum with the green crystal, and its minimum with the blue crystal. The two minimum in the data, the one shared by the red and blue crystal and the separate minimum of the green crystal, have similar peak temperature values of 1008.80 and 1008.83, respectively. However, the red and green crystals reach a much higher maximum value of 2200.12 than the blue crystal which tops out at 1401.76. As we can see from the histograms in Figure 67, the blue crystal is composed entirely of success scenarios which helps to explain its relatively low maximum temperature. Returning to the right image in Figure 66, we see the maximum point of the blue crystal is characterized by a high FailureTimeDG, ADSActivationTimeDelay, and DCPowerLife, and a low SRVStuckOpenTime compared to its adjoining red crystal. The green crystal separates itself from the red crystal at its minimum by having a low FWTime and high ACPowerRecoveryTime.

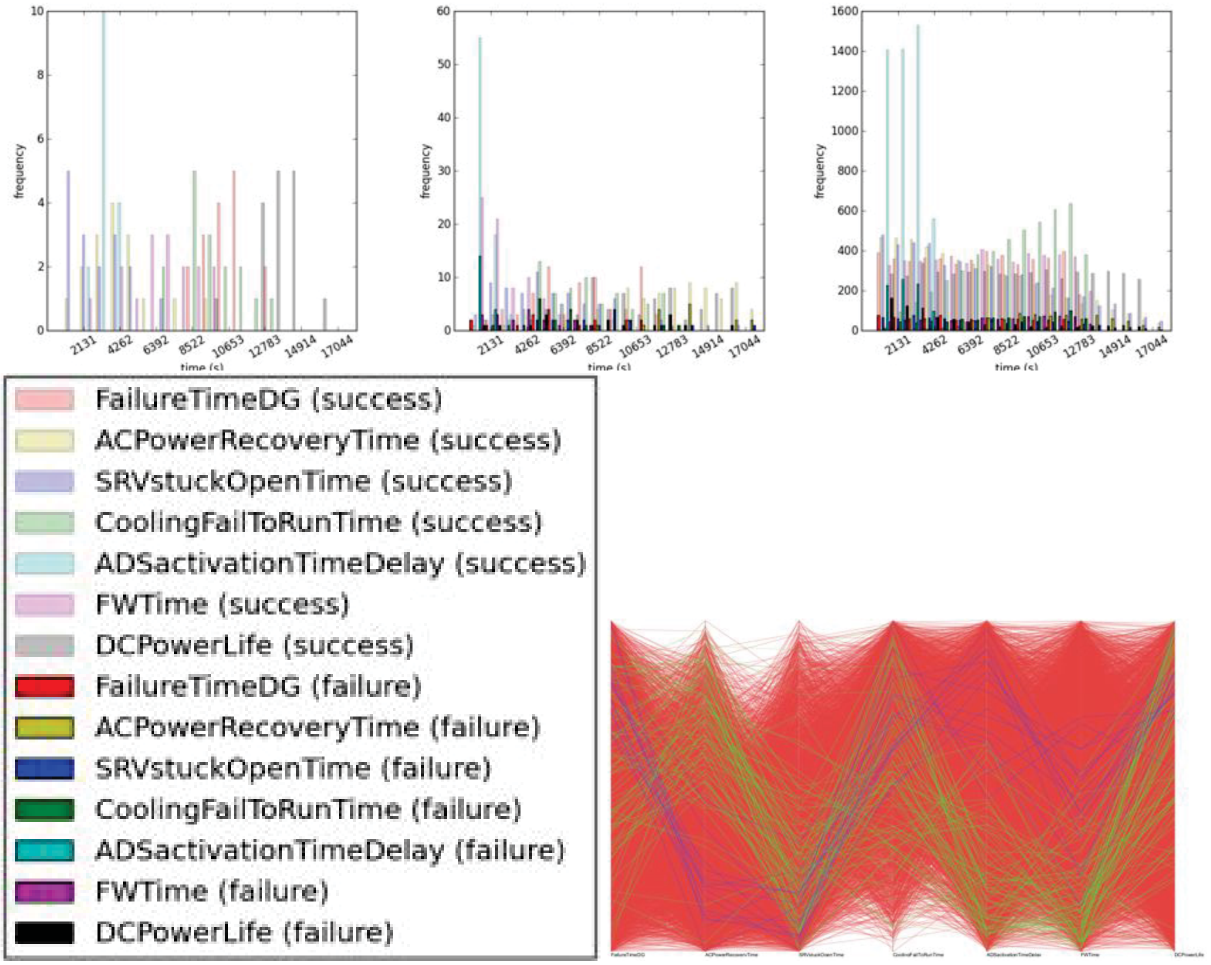


Figure 67: **7D-MT-All-3C**: distribution of input parameters across different clusters. Top row: blue, green and red clusters respectively. Bottom row: histogram legends for all 7 time-related input parameters for both failures and successes scenarios (left); parallel coordinate plots for all three clusters (right).

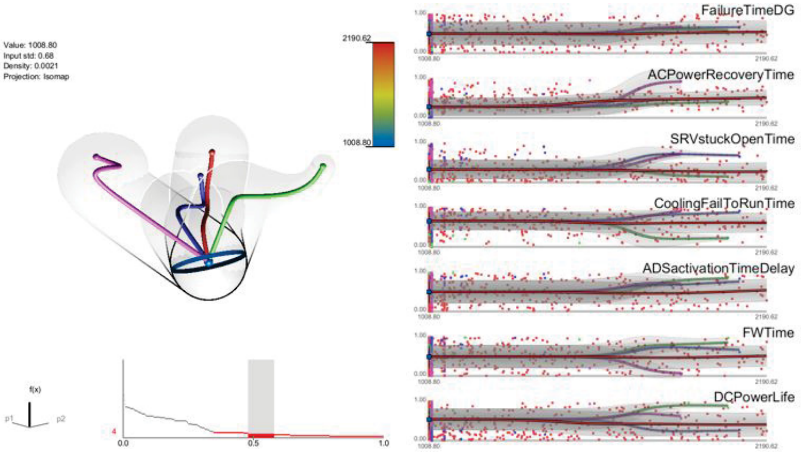


Figure 68: **7D-MT-Success-4C**: Left: Topological skeleton, right: inverse coordinate plots including summary regression curves.

6.5.5 7D-MT-Success-4C

As shown in Figure 68, there is one global minimum with a temperature value of 1008.80 connecting four crystals that arise to separate maxima with values of 1890.84 (magenta), 2056.14 (green), 2095.66 (blue), and 2190.68 (red). The majority of the 4164 success scenarios lie in the red crystal contains 3873 data points. The blue crystal has the next most points with 147 scenarios, followed by the green with 87, and lastly, the magenta which consists of 57 scenarios. Looking at the right image in Figure 68, we try to distinguish what is happening at each of the crystals' distinct maxima by focusing our attention on each input dimension. We notice that the regression curves in the FailureTimeDG dimension are very similar, whereas the DCPowerLife dimension showcases a good separation of the four crystals from each other. In this example, there are several input dimensions that separate the data into 3 distinct maximum modes: FWTime, where the blue and green crystals represent high value, the red crystal has a moderate value and the magenta crystal has a low value, and CoolingFailToRunTime, where the blue and magenta crystals represent high values, red again represents moderate value, and green represents low value. Three more dimensions split the data into two major modes, ADSActivationTimeDelay, SRVStuckOpenTime, and ACPowerRecovery. In the SRVStuckOpenTime case, the red and green crystals are similar as are the blue and magenta crystals. ADSActivationTimeDelay and the ACPowerRecovery distinguish one crystal from the other three. The red crystal has a lower value for ADSActivationTimeDelay than the other crystals. In ACPowerRecoveryTime, we see the magenta crystal has a much higher value than the other three. The distribution of input parameters across different crystals are shown in Figure 69.

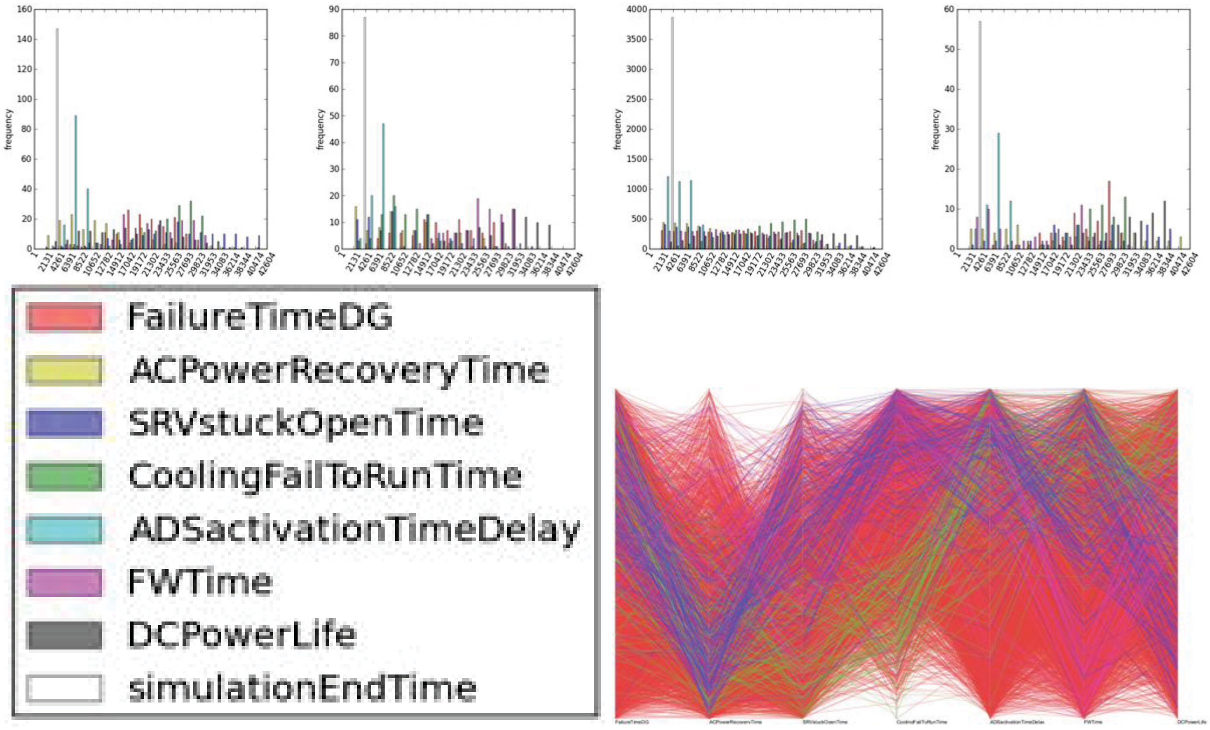


Figure 69: **7D-MT-Success-4C**: distribution of input parameters across different clusters. Top row: blue, green, red, and magenta clusters, respectively. Bottom row: histogram legends for all 7 time-related input parameters for both failures and successes scenarios (left); parallel coordinate plots for all four clusters (right).

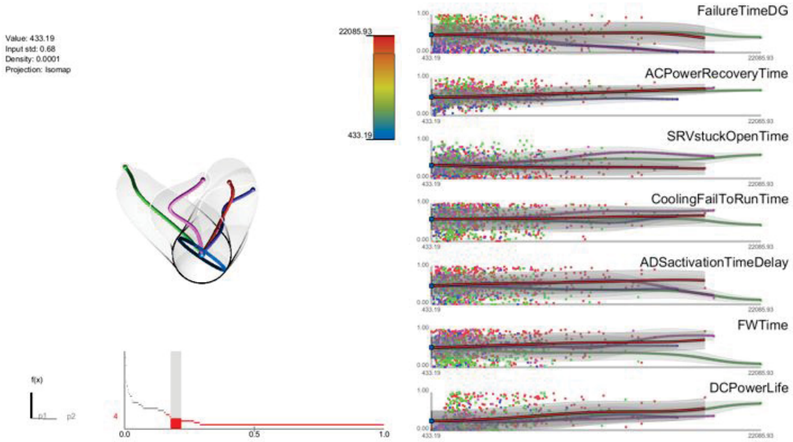


Figure 70: **7D-EST-Failure-4C**: Left: Topological skeleton, right: inverse coordinate plots with regression curves superimposed.

6.5.6 7D-EST-Failure-4C

For the 833 failure cases, we look at the end simulation time (EST) as the output variable, since the maximum temperature is the same for all of these scenarios. In this case, the clustering structure contains a single global minimum with EST valued at 433.19 (in seconds). The four distinct local maxima have values: 16580.71 (blue), 18396.70 (red), 18985.12 (magenta), and 22085.93 (green). In this setting if we focus on the coordinate plots in the right image of Figure 70, no single dimension can be used to reliably separate the four distinct maximum. We see that CoolingFailToRunTime, DCPowerLife, and ACPowerRecovery more or less exhibit the same behavior across each crystal. FWTime is able to distinguish the green crystal as having a distinctly lower value than the other three, whereas the ADSActivationTimeDelay shows the red crystal having a significantly higher value than the other three crystals. SRVStuckOpenTime shows the magenta and green crystals having higher values than the blue and red crystals. Lastly, the FailureTimeDG shows the red and green crystals having higher values than the blue and magenta crystals. The distribution of input parameters across different crystals are shown in Figure 71.

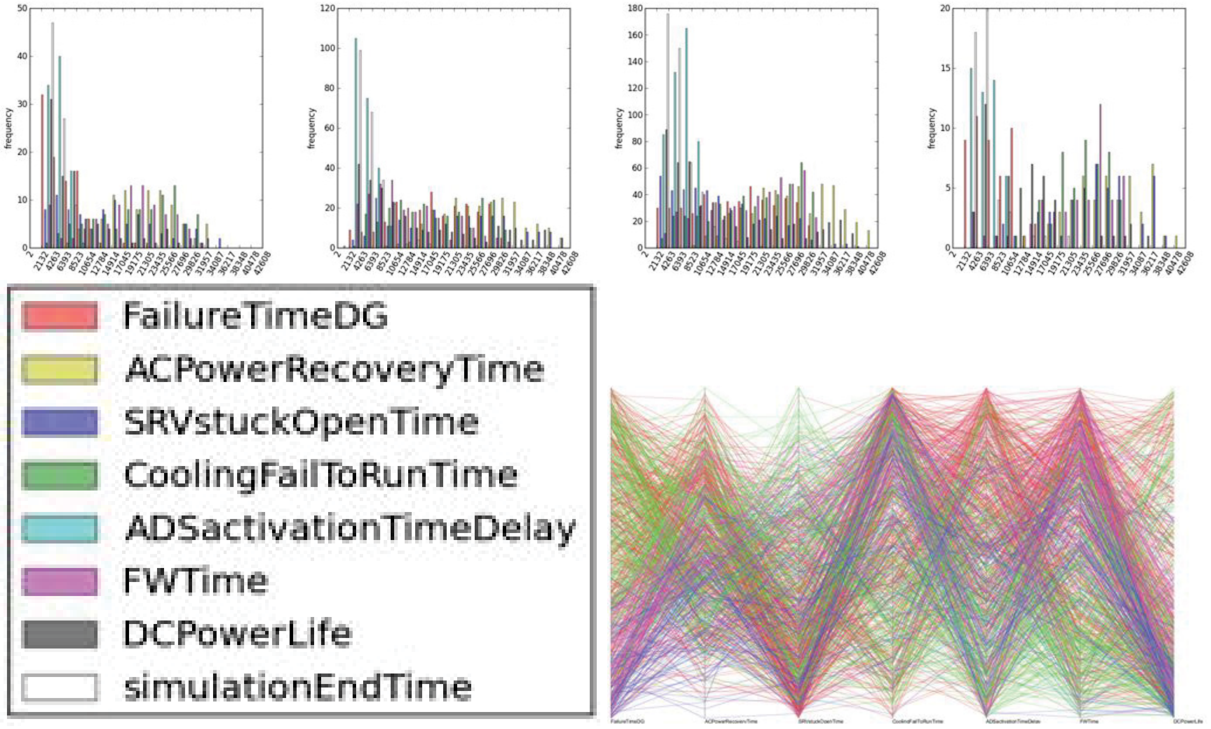


Figure 71: **7D-EST-Failure-4C**: distribution of input parameters across different clusters. Top row: blue, green, red, and magenta clusters, respectively. Bottom row: histogram legends for all 7 time-related input parameters for both failures and successes scenarios (left); parallel coordinate plots for all four clusters (right).

6.6 Discussions

In this section, we investigate the use of a topology-based clustering technique on data generated for a BWR SBO case study. Our clustering technique, inspired by the Morse-Smale complex which partitions the data based on their uniform gradient behavior, could offer complementary views regarding the topological structure of the data, in addition to traditional analysis. We demonstrate our analysis technique and visual interface by modeling the dataset as several high-dimensional scalar functions. We describe our initial observations regarding the various clustering results. We expect further collaboration with nuclear scientists to obtain in-depth understandings of correlations among various simulation parameters, to help with the design and simulation process.

7 Conclusions

When compared to the state-of-practice PRA tools, the major challenges in using Dynamic methodologies are (a) the heavier computational cost and (b) the large amount of data that needs to be analyzed. This project aims to improve furthermore Dynamic PRA methodologies by tackling these challenges using: (a) adaptive sampling techniques to reduce computational cost of the analysis and (b) topology-based methodologies to interactively analyze and visualize high-dimensional data and extract risk-informed insights.

First, we employ topology-based adaptive sampling algorithms that aim to infer or predict simulation outcome and decide the coordinates in the input space of the next sample that maximize the amount of information that can be gained from it. Such methodologies can be used to both explore and exploit the input space. The later one is especially used for safety analysis scopes to focus samples along the limit surface, i.e. the boundaries in the input space between system failure and system success.

Second, we present a software tool that is designed to analyze and visualize high-dimensional data, in particular, the datasets arise from nuclear simulations. Our tool includes a host of various analysis and visualization capabilities. We first perform topological analysis to identify the underlying topological structure of the dataset. Statistical information is then summarized and linked to the topological structures extracted from the data, allowing the users to identify correlations between timing/sequencing of events and simulation outcome. From an uncertainty quantification point of view, such a tool allows the user to identify and consequently rank variables based on their correlations with simulation outcome. Such software becomes even more effective for the analysis of complex systems such as nuclear power plants where the number of variables is very large.

The new generation of safety analysis codes (e.g., RELAP-7) will incorporate a new series of algorithms and dynamic probabilistic safety assessment methodologies (e.g., RAVEN) that are able to model system dynamics, model human interaction and digital control systems, and perform uncertainty quantification and sensitivity analysis. An objective in this trend is to use a best estimate plus uncertainty approach to nuclear reactor safety analysis instead of the traditional conservative approach. This process may result in large amounts of data generated that can be difficult to analyze and, from a user point of view, might be difficult to assess with regard to the main contributors to risk and most relevant trends. This project applies cutting edge methodologies and algorithms to safety analysis codes and data for nuclear systems in order to: analyze the simulations generated by generic safety analysis codes in an intuitive graphical form; identify correlations between sequence and timing of events and identify system characteristics and limitations using adaptive sampling techniques. These new techniques for visual analysis of uncertainty will provide a unique capability that will help safety analysts to more fully explore the high-dimensional landscape of uncertainties in a simulation. Moreover, we also focus on the development of adaptive sampling techniques that will help in exploring failure regions of the simulation without oversampling the space and, thus, greatly reduce the computational time of such analyses. Projects directly impacted include the Consortium for Advanced Simulation of Light Water Reactors (CASL) and NEAMS. The deliverables could be directly employed within existing PRA/UQ codes (i.e., RAVEN) in order to satisfy the needs of projects such as LWRS (in particular RISMC) campaign.

8 Project-Related Research Activities

We summarize research activities involving high-dimensional data analysis and visualization, in particular, applying to nuclear simulation datasets, conducted under the project, including publications, manuscripts and workshops.

Publications

- Exploration of High-Dimensional Scalar Function for Nuclear Reactor Safety Analysis and Visualization. Dan Maljovec, Bei Wang, Valerio Pascucci, Peer-Timo Bremer, Michael Pernice, Diego Mandelli and Robert Nourgaliev. Proceedings International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering (M&C) , pages 712-723, 2013.
- Adaptive Sampling Algorithms for Probabilistic Risk Assessment of Nuclear Simulations. Dan Maljovec, Bei Wang, Diego Mandelli, Peer-Timo Bremer and Valerio Pascucci. International Topical Meeting on Probabilistic Safety Assessment and Analysis (PSA) (accepted), 2013.
- Analyze Dynamic Probabilistic Risk Assessment Data through Topology-Based Clustering. Dan Maljovec, Bei Wang, Diego Mandelli, Peer-Timo Bremer and Valerio Pascucci. International Topical Meeting on Probabilistic Safety Assessment and Analysis (PSA) (Accepted), 2013.
- Adaptive Sampling with Topological Scores. Dan Maljovec, Bei Wang, Ana Kupresanin, Gardard Johannesson, Valerio Pascucci, Peer-Timo Bremer Working with Uncertainty Workshop at IEEE VisWeek, 2011. International Journal for Uncertainty Quantification , 3(2), pages 119-141, 2013.
- Branching and Circular Features in High Dimensional Data. Bei Wang, Brian Summa, Valerio Pascucci and Mikael Vejdemo-Johansson IEEE Transactions on Visualization and Computer Graphics , 17(12), pages 1902-1911, 2011.

Workshops

- SAMSI Uncertainty Quantification Transition Workshop, 2012 (Poster)
- Working with Uncertainty Workshop at IEEE VisWeek, 2011 (Presentation)
- Workshop on Topological Data Analysis and Visualization for Large-Scale and High-Dimensional Science Discover, during the International Topical Meeting on Probabilistic Safety Assessment and Analysis (PSA), 2013 (Organizer)
  Details: Massive scientific simulations, such as safety analyses, and sensing devices have generated large-scale and high-dimensional datasets which offer great challenges and opportunities for data analysis and visualization. In this workshop, we will give an introduction to

topological techniques which have become increasingly popular in data analysis. They come with provably robust computation, provide complete feature extraction and quantification, offer multi-scale structural descriptions with error-bounded approximations at each scale, and can be combined with streaming techniques to achieve scalable performance. The workshop will explore the field of topological data analysis and visualization with a particular focus on some recent applications in scientific domains such as nuclear safety simulations, material science, combustion and climate simulation. In the first part of the workshop, we will introduce our data management, analysis and visualization pipeline, highlight interplay between fundamental theory, robust algorithms, and their impact on applications. In the second part of the workshop, we will focus on exploration of high dimensional scalar functions, especially those arise from large-scale scientific simulations (e.g. simulation based safety analyses), with a live demo of our analysis and visualization tool.
Organizers: Bei Wang, Dan Maljovec, Diego Mandelli, Valerio Pascucci and Peer-Timo Bremer.
Presenters: Bei Wang, Dan Maljovec and Diego Mandelli.

Manuscript under Preparation

- Exploration of High-Dimensional Scalar Function for Nuclear Reactor Safety Analysis and Visualization. Dan Maljovec, Bei Wang, Valerio Pascucci, Peer-Timo Bremer, Michael Pernice, Diego Mandelli and Robert Nourgaliev. (Under preparation for the journal of Nuclear Engineering and Design).

- Adaptive Sampling Algorithms for Probabilistic Risk Assessment of Nuclear Simulations. Dan Maljovec, Bei Wang, Diego Mandelli, Peer-Timo Bremer and Valerio Pascucci. (Under preparation for submission to the journal of Reliability Engineering & System Safety).

- Analyze Dynamic Probabilistic Risk Assessment Data through Topology-Based Clustering. Dan Maljovec, Bei Wang, Diego Mandelli, Peer-Timo Bremer and Valerio Pascucci. (Under preparation for submission to the journal of Reliability Engineering & System Safety).

References

[1] C. Acosta and N. Siu. Dynamic event trees in accident sequence analysis: application to steam generator tube rupture. *Reliability Engineering and System Safety*, 41:135–154, 1993.

[2] A. Alfonsi, C. Rabiti, D. Mandelli, J.J. Cogliati, and R.A. Kinoshita. Raven as a tool for dynamic probabilistic risk assessment. In *Proceeding of American Nuclear Society (ANS), Atlanta (GA)*, volume 108, pages 555–558, 2013.

[3] Sunil Arya, David M. Mount, Nathan S. Netanyahu, Ruth Silverman, and Angela Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM*, 45:891–923, 1998.

[4] C. L. Bajaj, V. Pascucci, G. Rabbiolo, and D. Schikorc. Hypervolume visualization: a challenge in simplicity. *Proceedings IEEE Symposium on Volume Sisualization*, pages 95–102, 1998.

[5] Prosenjit Bose, Jean Cardinal, Sébastien Collette, Erik D. Demaine, Belén Palop, Perouz Taslakian, and Norbert Zeh. Relaxed gabriel graphs. *Proceedings Canadian Conference on Computational Geometry*, 2009.

[6] Peer-Timo Bremer, Herbert Edelsbrunner, Bernd Hamann, and Valerio Pascucci. A topological hierarchy for functions on triangulated surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 10(385-396), 2004.

[7] Russel E. Caflisch. Monte carlo and quasi-monte carlo methods. *Acta Numerica*, 7:1–49, 1998.

[8] Erik Carlsson, Gunnar Carlsson, and Vin de Silva. An algebraic topological method for feature identification. *International Journal of Computational Geometry and Applications*, 16:291–314, 2003.

[9] Gunnar Carlsson, Tigran Ishkhanov, Vin de Silva, and Afra Zomorodian. On the local behavior of spaces of natural images. *International Journal of Computer Vision*, 76:1–12, 2008.

[10] Gunnar Carlsson, Afra J. Zomorodian, Anne Collins, and Leonidas J. Guibas. Persistence barcodes for shapes. In *Proceedings Eurographs/ACM SIGGRAPH Symposium on Geometry Processing*, pages 124–135, 2004.

[11] Hamish Carr, Jack Snoeyink, and Michiel van de Panne. Simplifying flexible isosurfaces using local geometric measures. In *Proceedings 15th IEEE Visualization*, pages 497–504, 2004.

[12] D. I. Chanin, H. N. Jow, and J. A. Rollstin. *MELCOR Accident Consequence Code System(MACCS)*. Sandia National Laboratories, NUREG/CR-4691, Vols. 1-3, SAND86-1562, 1990.

[13] Frédéric Chazal, David Cohen-Steiner, Marc Glisse, Leonidas J. Guibas, and Steve Y. Oudot. Proximity of persistence modules and their diagrams. In *Proceedings 25th Annual Symposium on Computational Geometry*, pages 237–246, 2009.

[14] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete and Computational Geometry*, 37:103–120, 2007.

[15] Kree Cole-McLaughlin, Herbert Edelsbrunner, John Harer, Vijay Natarajan, and Valerio Pascucci. Loops in reeb graphs of 2-manifolds. In *Proceedings 19th Annual Symposium on Computational Geometry*, pages 344–350, 2003.

[16] Carlos D. Correa and Peter Lindstrom. Towards robust topology of sparsely sampled data. *IEEE Transactions on Visualization and Computer Graphics*, 17:1852–1861, 2011.

[17] L. Csató and M. Opper. Sparse online gaussian processes. *Neural Computation*, 14:641–668, 2002.

[18] Vin de Silva and Robert Ghrist. Coverage in sensor networks via persistent homology. *Algebraic and Geometric Topology*, 7:339–358, 2007.

[19] J. Devooght. Dynamic reliability. *Advances in Nuclear Science and Technology*, 25:215–278, 1997.

[20] Qiang Du, Vance Faber, and Max Gunzburger. Centroidal voronoi tesselations: Applications and algorithms. *SIAM Review*, 41(4):637–676, 1999.

[21] Herbert Edelsbrunner and John Harer. Persistent homology - a survey. *Contemporary Mathematics*, 453:257–282, 2008.

[22] Herbert Edelsbrunner, John Harer, Vijay Natarajan, and Valerio Pascucci. Morse-Smale complexes for piecewise linear 3-manifolds. *Proceedings 19th ACM Symposium on Computational Geometry*, pages 361–370, 2003.

[23] Herbert Edelsbrunner, John Harer, and Afra J. Zomorodian. Hierarchical Morse-Smale complexes for piecewise linear 2-manifolds. *Discrete and Computational Geometry*, 30:87–107, 2003.

[24] Herbert Edelsbrunner, David Letscher, and Afra J. Zomorodian. Topological persistence and simplification. *Discrete and Computational Geometry*, 28:511–533, 2002.

[25] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40, 1975.

[26] E. Melendez G. Cojazzi, J. M. Izquierdo and M. S. Perea. The reliability and safety assessment of protection systems by the use of dynamic event trees: the dylam-treta package. *Proceedings XVIII Annual Meeting Spanish Nuclear Society*, 1992.

[27] K. Gabriel and R. Sokal. A new statistical approach to geographic variation analysis. *Systematic Zoology*, 18:259–278, 1969.

[28] R. O. Gauntt. *MELCOR Computer Code Manual, Version 1.8.5, Vol. 2, Rev. 2*. Sandia National Laboratories, NUREG/CR-6119.

[29] Samuel Gerber, Peer-Timo Bremer, Valerio Pascucci, and Ross Whitaker. Visual exploration of high dimensional scalar functions. *IEEE Transactions on Visualization and Computer Graphics*, 16:1271–1280, 2010.

[30] John Geweke. Bayesian inference in econometric models using monte carlo integration. *Econometrica*, 57(6):1317–39, 1989.

[31] J. C. Gower. Measures of similarity, dissimilarity, and distance. In S. Kotz, N.L. Johnson, and C.B. Read, editors, *Encyclopedia of Statistical Sciences*, volume 5, pages 397–405. John Wiley & Sons, 1985.

[32] Attila Gyulassy, Vijay Natarajan, Valerio Pascucci, Peer Timo Bremer, and Bernd Hamann. Topology-based simplification for feature extraction from 3D scalar fields. In *Proceedings 16th IEEE Visualization*, pages 535–542, 2005.

[33] Attila Gyulassy, Vijay Natarajan, Valerio Pascucci, and Bernd Hamann. Efficient computation of Morse-Smale complexes for three-dimensional scalar functions. *IEEE Transactions on Visualization and Computer Graphics*, 13:1440–1447, 2007.

[34] A. Hakobyan, T. Aldemir, R. Denning, S. Dunagan, D. Kunsman, B. Rutt, and U. Catalyurek. Dynamic generation of accident progression event trees. *Nuclear Engineering and Design*, 238(12):3457–3467, 2008.

[35] W.K. Hastings. Monte carlo samping methods using markov chains and their applications. *Biometrika*, pages 97–109, 1970.

[36] J. C. Helton and F. J. Davis. Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. *Reliability Engineering & System Safety*, 81(1), 2003.

[37] E. Hofer, M. Kloos, B. Krzykacz-Hausmann, J. Peschke, and M. Woltereck. An approximate epistemic uncertainty analysis approach in the presence of epistemic and aleatory uncertainties. *Reliability Engineering and System Safety*, 77:229–238, 2002.

[38] K. S. Hsueh and A. Mosleh. The development and application of the accident dynamic simulator for dynamic probabilistic risk assessment of nuclear power plants. *Reliability Engineering and System Safety*, 52:297–314, 1996.

[39] Alfred Inselberg. *Parallel Coordinates: Visual Multidimensional Geometry and its Applications.* Springer, 2009.

[40] D. Jones, M. Schonlau, and W. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492, 1998.

[41] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: an Introduction to Cluster Analysis*. John Wiley and Sons, 1990.

[42] Jennifer Kloke and Gunnar Carlsson. Topological de-noising: strengthening the topological signal. Manuscript, 2010.

[43] C. Q. Lam. Sequential Adaptive Designs In Computer Experiments For Response Surface Model Fit. http://etd.ohiolink.edu/, 2008.

[44] Jun S. Liu. *Monte Carlo strategies in scientific computing*. Springer, 2001.

[45] E. Zio M. Marseguerra, J. Devooght, and P.E. Labeau. A concept paper on dynamic reliability via monte carlo simulation. *Mathematics and Computers in Simulation*, 47:371–382, 1998.

[46] David MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):589–603, 1992.

[47] F. Di Maio, M. Stasi, E. Zio, D. Mandelli, and T. Aldemir. Identification of faults in a level control dynamic system. In *Proceedings of NPIC-HMIT 2009, Knoxville (TN)*, 2009.

[48] Dan Maljovec, Avishek Saha, Peter Lindstrom, Peer-Timo Bremer, Bei Wang, Carlos Correa, and Valerio Pascucci. A comparative study of Morse complex approximation using different neighborhood graphs. Workshop on Topological Methods in Data Analysis and Visualization (accepted), 2013.

[49] Dan Maljovec, Bei Wang, Ana Kupresanin, Gardard Johannesson, Valerio Pascucci, and Peer-Timo Bremer. Adaptive sampling with topological scores. *International Journal for Uncertainty Quantification*, 3(2):119–141, 2013.

[50] Dan Maljovec, Bei Wang, Valerio Pascucci, Peer-Timo Bremer, Michael Pernice, Diego Mandelli, and Robert Nourgaliev. Exploration of high-dimensional scalar function for nuclear reactor safety analysis and visualization. *International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering*, 2013.

[51] D. Mandelli and C. Smith. Adaptive sampling using support vector machines. In *Proceeding of American Nuclear Society*, volume 107, pages 736–738, 2012.

[52] D. Mandelli, A. Yilmaz, and T. Aldemir. Data processing methodologies applied to dynamic PRA: an overview. In *Proceeding of Probabilistic Safety Analysis*, 2011.

[53] D. Mandelli, A. Yilmaz, and T. Aldemir. Scenario analysis and PRA: Overview and lessons learned. In *Proceedings of European Safety and Reliability Conference*, 2011.

[54] Diego Mandelli, Alper Yilmaz, Tunc Aldemir, Kyle Metzroth, and Richard Denning. Scenario clustering and dynamic probabilistic risk assessment. *Reliability Engineering & System Safety*, 115:146 – 160, 2013.

[55] Diego Mandelli, Alper Yilmaz, Kyle Metzroth, Tunc Aldemir, and Richard Denning. Scenario aggregation and analysis via mean-shift methodology. *Proceedings International Congress on Advances in Nuclear Power Plants*, 2010.

[56] G. W. Milligan and M. C. Cooper. A study of standardization of variables in cluster analysis. *Journal of Classification*, 5:181–204, 1988.

[57] John Milnor. *Morse Theory*. Princeton University Press, New Jersey, NY, USA, 1963.

[58] H. N. Najm. Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics. *Annual Review of Fluid Mechanics*, 41:35–52, 2009.

[59] Owen. Orthogonal arrays for computer experiments, integration and visualization. *Statistica Sinica*, 2:439–452, 1992.

[60] C. Rabiti, A. Alfonsi, D. Mandelli, J. Cogliati, and R. Martineau. Raven as control logic and probabilistic risk assessment driver for relap-7. In *Proceeding of American Nuclear Society (ANS), San Diego (CA)*, volume 107, pages 333–335, 2012.

[61] RELAP5-3D Code Development Team. Relap5-3d code manual, 2005.

[62] M. Schonlau. Computer Experiments and Global Optimization. PhD thesis, University of Woterloo, 1997.

[63] N. Siu. Risk assessment for dynamic systems: an overview. *Reliability Engineering and System Safety*, 43(1):43–73, 1994.

[64] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

[65] R. Winningham, K. Metzroth, T. Aldemir, and R. Denning. Aircraft crash recovery scenario dynamic event tree analysis of the RVACS passive decay heat system employing the ADAPT tool with RELAP5-3D. *Proceedings American Nuclear Society*, 2009.

[66] R. Winningham, K. Metzroth, T. Aldemir, and R. Denning. Passive heat removal system recovery following an aircraft crash using dynamic event tree analysis. *Proceedings American Nuclear Society*, 100:461–462, 2009.

[67] E. Zio. Reliability engineering: Old problems and new challenges. *Reliability Engineering and System Safety*, 94(2):125–141, 2009.

[68] E. Zio and F. D. Maio. Processing dynamic scenarios from a reliability analysis of a nuclear power plant digital instrumentation and control system. *Annals of Nuclear Energy*, 36:1386–1399, 2009.